AN INTERACTIVE DECISION SUPPORT SYSTEM

FOR SCHEDULING FIGHTER PILOT TRAINING

THESIS

Cuong T. Nguyen, Captain, USAF

AFIT/GOR/ENS/02-12

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
26 Mar 02	Final	May 2001 - Mar 2002

Title and Subtitle An Interactive Decision Support System for Scheduling Fighter Pilot Training	**Contract Number**
	Grant Number
	Program Element Number
Author(s) Capt Cuong T. Nguyen, USAF	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Bldg 640 WPAFB OH 45433-7765	**Performing Organization Report Number** AFIT/GOR/ENS/02-12
Sponsoring/Monitoring Agency Name(s) and Address(es) 87th Flying Training Squadron, Public Affairs 561 Liberty Drive., Suite 3, Laughlin AFB TX 78843-5227	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes The original document contains color images.	

Abstract

Fighter Pilots students undertake an intense 120-day training program. New classes of students enter the training program at regular interval. Students endured rigorous academic, simulator, and aircraft training throughout the program. Squadron schedulers ensure the multiple classes and students are scheduled for the activities. Simulator and aircraft training are scheduled individual for each student. Academic training are taught to the class. Aircraft utilization must also be considered. Aircraft Sortie training are also constrained by daylight hours. Additionally, students are limited to a maximum of three training events in a given day. Squadron schedulers must balance these requirements to ensure students meet their training requirements and successfully graduate. The dynamic training environment requires advanced robust schedules with flexibillity to accommodate changes. A Visual Interactive Modeling approch is used to generate schedules. Current schedules are being generated manually with an Excel spreadsheet. Taking advantage of Excel's Visual Basic Programming language, the Excel tool is modified in several ways. Scheduling Dispatch rules are implemented to automatically generate feasible schedules. Graphical User Interfaces are used to create a user-friendly environment. Schedulers guide the schedule building process to prodcue a robust schedule. An attrition environment is created to simulate attrition probailities of aircraft sortie training due to operations, maintenance, weather, and other cancellations. Analysis of dispatch rules are analyzed.

Subject Terms
Scheduling, Fighter Pilot Scheduling, Excel VBA Applications

Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU

Number of Pages
116

AFIT/GOR/ENS/02-12

AN INTERACTIVE DECISION SUPPORT SYSTEM

FOR SCHEDULING FIGHTER PILOT TRAINING

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Cuong T. Nguyen, B.S.

Captain, USAF

March 26, 2002

AFIT/GOR/ENS/02-12

AN INTERACTIVE DECISION SUPPORT SYSTEM

FOR SCHEDULING FIGHTER PILOT TRAINING

Cuong T. Nguyen, B.S.

Captain, USAF

Approved:

/s/_____   <u>26 March 2002</u>
       Dr. Richard F. Deckro (Advisor)   date

/s/_____   <u>26 March 2002</u>
       Dr. James W. Chrissis (Reader)   date

Acknowledgments

Table of Contents

List of Figures

List of Tables

<u>Abstract</u>

Fighter Pilot students undertake an intense 120-day training program. New classes of students enter the training program at regular intervals. Students endure rigorous academic, simulator, and aircraft training throughout the program. Squadron schedulers ensure that multiple resources and students are scheduled to facilitate these activities. This includes both the scheduling of entire flights as a group for classroom work, and individuals for simulator and flying sorties. In addition, regulations impose a number of restrictions. Squadron schedulers must balance these restrictions to ensure students meet their training requirements and graduate. The dynamic training environment requires a robust scheduling approach with flexibility to accommodate changes due to a number of factors.

A Visual Interactive Modeling approach is used to generate schedules. To facilitate acceptance, this model was extended for the current approach of manually generating a schedule with an Excel spreadsheet. Taking advantage of Excel's Visual Basic programming language, the Excel tool was modified in several ways. Scheduling dispatch rules are implemented to automatically generate feasible schedules. Graphical User Interfaces are used to create a user-friendly environment. Schedulers guide the schedule building process to produce a robust schedule. In addition to developing a scheduling tool, an attrition environment is created to simulate attrition of aircraft sortie training due to operations, maintenance, weather, and other cancellations. Analysis of dispatch rules is provided.

AN INTERACTIVE DECISION SUPPORT SYSTEM

FOR SCHEDULING FIGHTER PILOT TRAINING

Chapter I: Introduction

Background

The United States Air Force (USAF) is the most advanced air force in the world
today. State-of-the-art aircraft such as the F-22 Raptor, F-117 Nighthawk, B-1B Lancer
and the B-2 Spirit are at the leading-edge of aircraft technology. The pilots flying these
aircraft are among the most highly skilled in the world. Each year, the USAF's pilot
training program turns out approximately 160 new fighter pilots at Laughlin Air Force
Base (AFB), one of three fighter pilots training bases in the United States. Entry into the
US's extremely successful training programs is highly sought after by other countries.
Our allies send many of their most promising pilots to the US to take undergraduate pilot
training at Laughlin AFB and other bases (McCurdy, Interview).

Laughlin AFB is the largest of three Undergraduate Pilot Training (UPT) schools.
In addition to Laughlin AFB, Columbus AFB in Mississippi and Vance AFB in
Oklahoma also train pilots for various missions (AETC Syllabus, p.1). The 87th Flying
Training Squadron (FTS), using the T-38 Talon fighter trainer, graduate fighter pilots for
the U.S and our allied Air Forces (87th FTS OI, 2000). Approximately 180student pilots
go through the fighter pilot training program each fiscal year. With such a high student

throughput rate, the 87th FTS is always pursuing ways to improve the training while better utilizing their resources.

One such potential area to assist the 87th is in the scheduling of the training program. As manning and resources evolve, organizations have to move people from position to position to cover the rotation of personnel. In most squadrons, whether training or operational, a loss of experienced schedulers, through rotation or reassignment, equates to loss of critical knowledge, which often must be "re-invented" and learned by the new officer who is charged with scheduling.

At the 87th FTS, one of the instructor pilots (IPs) is designated as the squadron scheduler. Scheduling is an additional duty for the IP beyond his instructor responsibilities. Due to operational and other requirements, the 87th FTS rotate their IP/Scheduler about once every six months (87th FTS OI, 2000). While expert pilots, new schedulers often have limited experience in the science of scheduling. Such situations often lead to a reduction in the scheduler's productivity in the first few months as they learn the details of scheduling the squadron and gain experience in the scheduling process (Calhoun, Interview). The schedule is generated, but it often requires more of the officer's time than would have been taken by an experienced scheduler. Any improvement in the scheduling process that can assist the new schedulers in decreasing the time required to learn the process. In addition, it would generate new schedules that will reduce the scheduler's workload while offering the potential of improved squadron schedules. In addition to directly impacting the scheduler's time, improvement in the initial schedule will reduce the need for re-scheduling "negotiations" throughout the

squadron, leaving more time for instructors to focus on their primary duties as IPs. The

reduction of re-scheduling can also have a beneficial effect on aircraft utilization.

The squadron schedule determines when a student pilot attends classes, train in

the simulator, and fly sorties in the T-38. The squadron schedule thus directly affects all

training processes for the student pilots. An improved schedule provides an opportunity

for a better training process and can potentially lead to graduating more fighter pilots.

Problem Statement

The current scheduling problems encountered at the 87th FTS can be broken

down into four main areas. The scheduling process is *ad hoc*, with no fully automated

ability to re-schedule once the process starts. If there are any changes to the schedule,

they must be dealt with manually as they occur. The flying squadron also draws IP's

from the reserve units, but does not have a direct call on the reservists' services.

Requests must go through the reservist office's POC. Because of the variations in the

scheduling process, the manpower and resources required to maintain the aircraft can

potentially be utilized inefficiently.

A training squadron encounters more dynamic changes to its schedule than a

normal fighter squadron. The 87th FTS schedules about 82 sorties per day (based on

their current schedule). With a large population of student and instructor pilots, more

sorties are often flown per day in a training squadron than a normal fighter squadron

during peacetime. Because this is a learning environment, students are less experienced

and are more prone to "bust a ride," forcing the need to re-schedule subsequent sorties,

either to utilize a sortie for which the original student did not qualify to take or to add

additional training for other students. Weather may also affect flight scheduling,

particularly when the student pilots are not yet qualified for the conditions or if the

conditions are too dangerous for inexperienced pilots. If the weather changes (visibility

decreases, or temperature is too high, for example), some sorties have to be re-scheduled

to a later time or cancelled for the day. Seasonal factors such as extreme heat during a

summer afternoon, or foggy weather during the morning hours, must also be considered

when building a schedule.

The squadron scheduling process consists of several steps. The flights initiate the

scheduling process by submitting the flight request for the weekly schedule. The

squadron scheduler receives the flights' requests and coordinates with maintenance to

determine required resources for the weekly schedule. The squadron scheduler assigns

the flight order with details of time and sorties available to the flight schedulers. The

flight schedulers received the draft schedule and, in turn, fill out the details of individual

sorties for students and instructor pilots (IPs) according to the available time slot. Once

the flight scheduling details are finished, the result is returned to the squadron scheduler

who updates the schedule. There is no formal system of communication between the

flight schedulers in the scheduling process.

As daily changes occur, the flight and squadron schedulers make changes "on the

fly" as circumstances require. If a flight is cancelled, they will try to find student pilots

or instructor pilots to fill the opening in the schedule using "task-by-line-of-sight". This

potentially results in some instructors being over-tasked, while others may have limited

flying time for a particular day. While these last minute adjustments will always occur,

decision support tools can assist in the re-scheduling effort by providing the scheduler with a list of students and instructors who are available for re-scheduling of sorties.

Currently the 87th FTS flies about 82 sorties per day; each of these sorties requires maintenance crew to prep the aircraft for take-off. The more sorties flown each day, the more maintenance crew time is required. The impact of any schedule on maintenance must be considered.

Scope

Scheduling is a common problem in the Armed Services and in the civilian sector. Schedules can be affected by a multitude of factors, both tangible and intangible. The scope of this thesis is to provide a decision support system to assist the 87th FTS resolve problems associated with scheduling and re-scheduling flight training schedules, and to maximize sorties while meeting training requirements.

Methodologies

To assist the 87th FTS, a Squadron Scheduling Decision Support Tool (SSDST) is developed. The SSDST is spreadsheet-based scheduling software, and has been developed from a framework of the current tool being utilized at the 87th. This framework was used because of the squadron's familiarity with the current spreadsheet method. Modifications of the spreadsheet include adding a scheduling engine and making the software user-friendly.

A key to the SSDST scheduler is the development of a user-friendly system and interface. Ease of use is assisted by user-friendly "buttons" or Grapical User Interface

(GUI). Codes exists behind the GUIs; pushing a button activates the codes to perform the scheduling functions. The resulting initial schedules generated from the SSDST allows over-rides and modifications to the schedules to give the commanders and schedulers desired flexibility. Final schedules and changes can be saved as separate files for portability.

The rest of the thesis is organized as follows: Chapter II covers the background to the scheduling problem and the relevant literatures that guide the direction of the thesis. Chapter III develops the methodology to define the framework of the software, the test environment, and the test scenario. Chapter IV covers the implementation of the scheduling dispatch rules and the analysis of the test scenario. Chapter V summarizes the results of the analysis and suggests recommendations for future research.

Chapter II: Concept Definition and Literature Review

General

  This chapter provides the background of the scheduling environment at the 87th FTS. The chapter covers the concepts of scheduling, scheduling in the pilot training environment, pertinent literature on multi-criteria schedules, and using software based visual interactive modeling to generate schedules. In addition, it mentions both the current tool used to generate schedules and the new training management tool currently being installed at the training bases.

Introduction

  "Scheduling concerns the allocation of limited resources to tasks over time. It is a decision making process that has as a goal the optimization of one or more objectives." [Pinedo, p.1]. Scheduling is a decision-making process that exists in almost all operational environments. A manufacturing facility has to manage the flow of its resources: the arrival of raw material, worker shifts, and departure of finished products. A "soccer mom" has to juggle shopping for grocery, picking children up from school, cleaning the house, picking up the dry cleaning, taking the dog to the veterinarian, and a host of other tasks. In general, scheduling is the problem of sequencing a set of jobs and allocating them to certain time slots without violating certain constraints [Klein, et al,].

Scheduling Problem

  The scheduling problem has attracted much interest from both academia and the operational world [Evren, 1999]. Many theoretical research topics are directed towards

simple machine scheduling problems. In the operational world, scheduling environments are much more complex and cannot be directly extrapolated to some simple theoretical machine-scheduling model. Pinedo outlines some of the most common scheduling problems encountered in practice. Empirically, scheduling problems that are relevant to resource scheduling environments may be summarized as:

> Theoretical models usually assume that there are n jobs to be scheduled and that after scheduling these n jobs the problem is solved. In reality, new jobs are added or current jobs are re-scheduled continuously. The dynamic nature of resource scheduling in services may require that slack times be built into the schedule in expectation of the unexpected.

> Theoretical models usually do not emphasize the re-sequencing problem. In practice, some random event may require major changes and the reactive scheduling (re-scheduling) process, may have to satisfy certain constraints. Thus, stochastic scheduling environments, might benefit from robust schedules in lieu of some optimality objective.

> Real world scheduling environments are often more complicated than the ones considered in general scheduling theory.

> In the mathematical models, the weights (priorities) of the jobs are assumed to be fixed, that is they do not change over time. In practice, the weight of a job often fluctuates over time due to changing priorities in the organization, different goals being emphasized, or a number of other factors.

> Mathematical models often do not take preferences into account. A scheduler may favor some assignment for some reasons that cannot be incorporated into the model.

> Most theoretical research has focused on models with a single objective. Most real world problems exhibit multi-criteria and multi-objective characteristics, which sometimes are in conflict with each other. [Pinedo, 1995]

Pinedo states that scheduling is the decision-making process that exists in most manufacturing and production systems as well as in most information-processing

environments [Pinedo, 1995, p. 1]. Scheduling in these settings allocates resources to different tasks over a period of time. The resources and tasks may be in different forms. Resources may be machines in a workshop or runways at an airport, and tasks may be operations in a production system or take-offs and landings at an airport [Pinedo, 1995, p. 1].

In the next section, scheduling in the fighter training squadron is discussed. It is important to understand the critical factors in the scheduling environment to assist in understanding the focus of the scheduling process in this thesis. The critical factors are discussed in the section; more details on various factors can be found in Appendix A.

Scheduling in the 87th FTS Environment

The 87th Flying Training Squadron (FTS) is a graduate training squadron training future jet fighter pilots. Its main mission is to train graduates from undergraduate pilot training programs, readying pilots to fly fighter jet aircraft [McCurdy, interview; and 87th FTS OI, 2000]. The 87th FTS is staffed by both experienced pilots from the field and newly-graduated, less experienced, instructors graduating near the top of the previous class. These superior students are retained to become Instructor Pilots (IP), training new pilots and gaining valuable experience and flight hours. The flight training cycle at the 87th FTS lasts approximately 6 months. Students come from undergraduate pilot training (UPT) from the 84th FTS (also located at Laughlin AFB), from other UPT bases (Columbus AFB, MS and Vance AFB, OK, and others [AETC Syllabus, p.1]), and from allied countries in many parts of the world [McCurdy, interview]. Each class starts approximately three weeks behind the previous class. Each of the four flights making up

the 87th FTS has two classes, a senior and junior class. Therefore, at any given time, eight classes at different levels of training share the squadron's resources [McCurdy, interview]. An entire training period includes a sequence of precedence-related events such as orientation, academics, avionics/cockpit familiarization training (AFT/CFT), pre-flight simulator sorties, and flight training [87th FTS OI]. The four flights in the squadron are assigned IPs according to the IPs' specialties. The IPs train the students within their assigned flight, but may train students in other flights as situations require.

In addition, the maintenance shop services the 55 aircraft assigned to the 87th FTS. Some of these aircraft are two-seat (tandem) trainer models, while the rest are single-seat models. Most are operational at any given time. However, downtime does occur for repair, maintenance, or qualifying checks [McCurdy, interview 2001].

The Scheduling Shop

Flight training may be characterized as being similar to a manufacturing operation. Pinedo gives an example of a manufacturing system [Pinedo, p. 3]. In the example, the manufacturing system processes job orders with due dates. The jobs utilize resources such as machines and workspaces. Detailed scheduling of the tasks performed in the production system is necessary to maintain efficiency and effective control of operations. The production system also encounters unexpected events that have an impact on the scheduling. Unexpected breakdowns of machines or processing times that are longer than anticipated may have a significant impact on the overall schedule.

The scheduling process at the 87th FTS exhibits similar characteristics to the manufacturing system. Job orders, classes, simulator sorties, and flying sorties all have

10

processing times and due dates. Resources such as the T-38 training aircraft and Instructor Pilots are non-depleted resources. Detailed scheduling of the students, sorties, and classes is necessary to ensure all required students are available for assigned class lectures, simulator training, and training sorties. Bad weather and aircraft breakdowns unexpectedly occur, reducing the amount of time available to fly training sorties. The raining requirement to meet the minimum sortie training hours may be adversely impacted by these unexpected events.
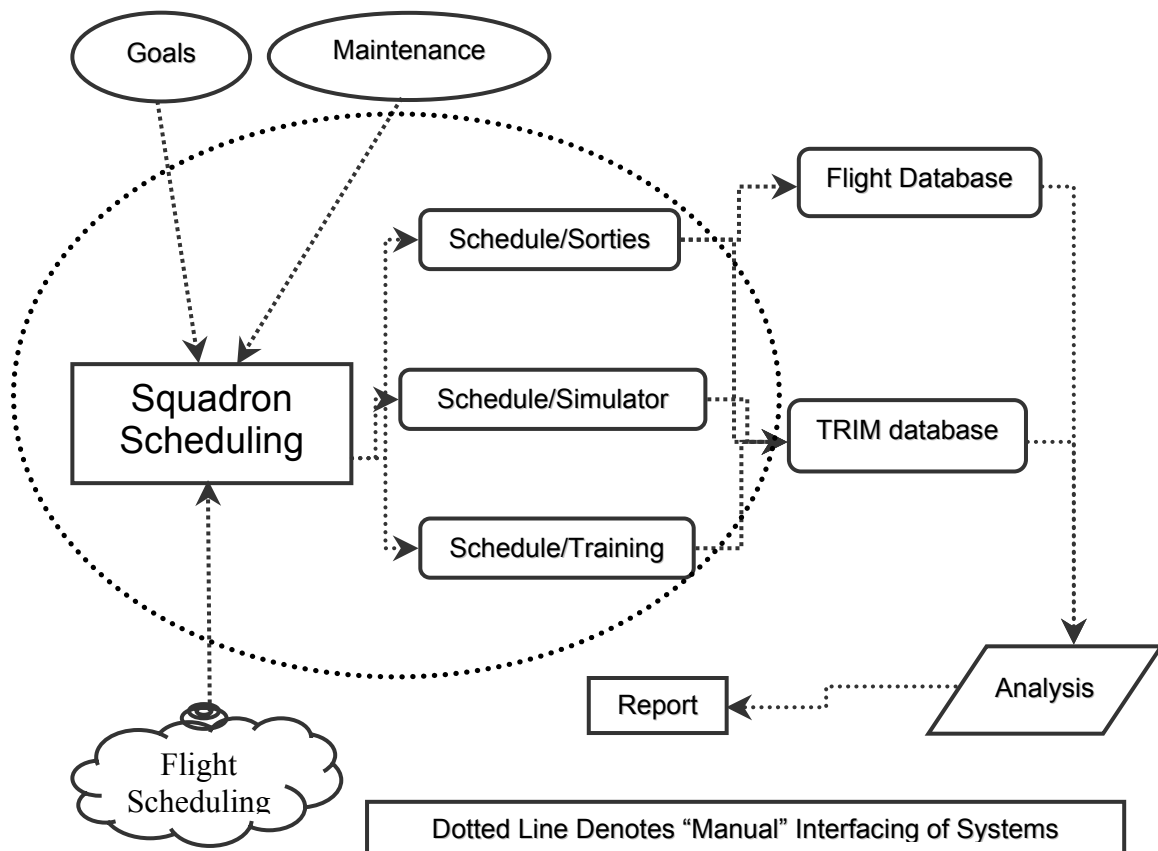
Figure 1. The Squadron Scheduling Process

Squadron scheduling directly influences every pilot's life in the squadron. The schedule determines when the pilots attend classes, when they fly sorties in the simulator pods, or when they fly the aircraft to gain valuable skills and experience. Without a schedule, how would they know when to fly, whom to fly with, and what plane they are to flying in? [McCurdy, interview]

To begin examining the squadron scheduling process, a high-level look at the process is needed. Figure 1 is a process flowchart representing the squadron scheduling process. It is compiled from descriptions of the process through interview with the 87th FTS scheduler, [McCurdy, interview] and the AETC Training Syllabus [AETC Syllabus]. The main outputs of the squadron scheduling shop are the weekly schedules [Calhoun, interview]. Generating a weekly draft schedule is the starting point for the weekly squadron scheduling. In Figure 1, the cloud represents the four flights submitting their sorties, simulator, and classroom requests. Information about availability of aircraft are taken from the maintenance squadron. Classrooms must also be available for class lectures. Information about the squadron's long-range plans are also required for planning a weekly schedule generation cycle [Calhoun, interview]. Long-range goals and monthly goals for flying hours are used to determine whether a flight or student requires additional sortie training or receives a higher priority when determining schedules. Information from internal and external sources are used to generate the draft schedule. Upon completion of the draft schedule, the maintenance department matches available aircraft to the proposed schedule. The draft schedule is then checked for feasibility and modified if necessary until satisfactory schedules are reached. The outputs of the scheduling process are the three different schedules: sortie schedule, simulator

schedule, and classroom schedule. The final schedules are relayed to appropriate external

entities such as maintenance, flight dispatching, tower, and others [Calhoun, interview].

The Scheduling Process

To graduate from the fighter pilot training program students must meet minimum

hours in both simulator training and flying sorties. In addition to these hours, students

must also pass all exams from both the class lectures and the independent computer-aided

instruction (CAI). To meet these requirements, the squadron scheduling shop develops

draft schedules for class lectures, simulator sorties, and flying sorties. Class lectures are

required to introduce students to appropriate concepts. Once the students learn the

classroom material and pass examination, the student attends simulator training for the

appropriate material. After simulator training is passed, the student may begin flying

sorties for the appropriate material already covered.

The main goals for the squadron's scheduling effort are to meet certain sorties or

flying hour goals, balancing the requirements of the four different flights, and

maintaining a balance in the students' class work and simulator training. The squadron

scheduler prepares the schedules to meet the flying goals based on the availability of the

input factors: air traffic controller, students, IPs, weather, night sorties, prerequisites, and

other factors. Figure 2 represents the squadron-level scheduling input sources.
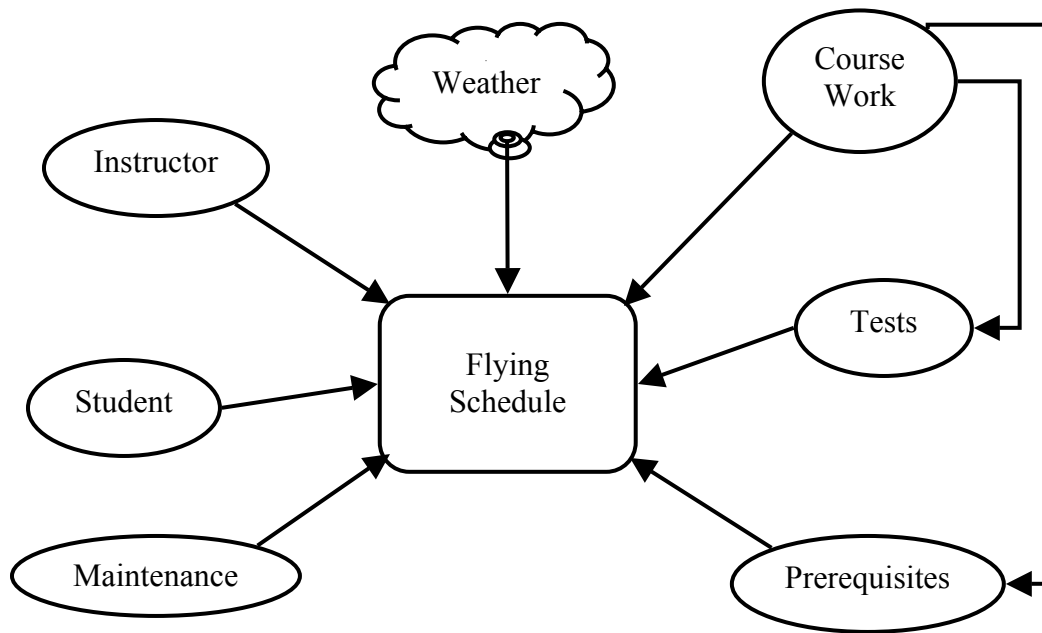
Figure 2. Squadron Scheduling Input Sources

Squadron scheduling must consider minimum hourly goals/requirements of the students. In order for the students to graduate, all students must successfully complete training hours of: at least 29.9 hours of simulator, 117 out of 118.7 hours of T-38 aircraft, and approximately 247.3 hours of ground training [AETC Syllabus, pg:1-2]. Ground training includes in-class academics, physical training, and individual self training. Students who do not meet the requirements are washed back to the next junior class to complete requirements. A student must pass the appropriate classroom and simulation training before the student can fly the training sortie in a particular phase of training. These requirements for a sequence of training phase force precedence relations on requirements.

Meeting the students' hourly goals requires three different schedules. One of these is the *academic schedule*, scheduling classrooms for in-class lectures and

independent CAI. This schedule requires all students in the same class to be available.

In-class academic education and CAI introduces the students to the fundamentals of the

curriculum objective. Students must successfully pass the lecture and the lecture exam to

demonstrate competency in the material. Once the lecture and CAI exams are completed,

the student proceeds to simulator training.

Simulator scheduling provides students with cockpit environment training without

leaving the ground. Students learn and practice initial flying skills of previously-covered

materials in a simulated flying environment. Simulator sorties require exactly the same

procedures and time requirements as aircraft sortie training. Students must demonstrate

proficiency in simulator training before proceeding to aircraft sortie training.

Aircraft sortie scheduling allocates students and IPs to available aircraft to take

off at assigned intervals throughout the day. Students (and accompanying IPs if required)

train in the T-38 aircraft. The schedule allocates students to take off at certain times at

the runway. Time-slot allocations are five minutes between aircraft take-offs.

The squadron scheduling shop receives the appropriate information to generate

these schedules from various sources. The flights submit the classroom, simulator, and

sorties requests. The squadron scheduler checks the students' and flights' training

progress. The number of daylight hours is also an input. The squadron scheduler also

considers maintenance's aircraft availability. From this information, the squadron

scheduler creates the schedules for classrooms, simulator training, and aircraft sorties.

One of the areas of interest is how to generate a good schedule in the face of

varied and sometimes conflicting objectives. Students want to maximize their training

hours, flying as much as possible. Maintenance wishes to minimize the number of

unused aircraft to prepare to minimize on cost of time and manpower. Sometimes sorties

must be cancelled or delayed due to outside or uncontrollable factors such as visiting

dignitaries, weather, failure to meet required prerequisites for the training sortie, or other

factors. Any scheduling approach adopted must consider how to balance these

competing objectives.

Multiple Criteria Optimization

Multiple criteria optimization is a technique from the field of multiple criteria

decision making (MCDM) [Steuer, p. 5]. Multiple criteria optimization utilizes

mathematical programming to analyze problems with multiple, and sometimes

conflicting, objectives to arrive at a mathematically optimal solution. Steuer states that

"… a problem has multiple objectives when it possesses multiple conflicting criteria."

[Steuer, p.vii]

The analysis of a multiple criteria problem begins by formulating the problem as a

multiple objective linear program (MOLP). Steuer formulates the MOLP as follows:

[Steuer, p.138]

$$\max\{\ c^1 x = z_1\}$$
$$\max\{\ c^2 x = z_2\}$$
$$\vdots$$
$$\max\{\ c^k x = z_k\}$$
$$s.t. \quad x \in S$$

where:
k : the number of objectives.
C^i : the gradient (vector of objective function coefficients) of the ith
            objective function.

Z_i : the criterion value (objective function value, z-value) of the ith objective
S : the feasible region.
max   : indicates that the purpose is to maximize all objectives simultaneously.
C : the k x n criterion matrix (matrix of objective function coefficients) whose rows are the gradients C^i of the k objective functions.
z : the criterion vector (objective function vector, z-vector).

Multiple objective problems rarely have points that simultaneously maximize all of the objectives. The solution is obtained by maximize each of the objectives to the "greatest extent possible" [Steuer, pp. 138-139].

Some recent research efforts in the multiple criteria scheduling area are worth noting. Klamroth and Wiecek examine scheduling production on a single machine using a dynamic programming (DP) based algorithm [Klamroth and Weicek, p.17] Klamroth and Weicek propose a DP approach to solve a time-dependent multiple criteria scheduling problem. The problem deals with scheduling time-dependent jobs or projects to be completed on a single machine. The model uses a continuous-time variable with each job completion yielding specified benefits. The schedule is defined as feasible when the generated schedule of jobs does not exceed the machine's capacity. The benefit of the schedule is calculated as the sum of the benefits of all jobs in the schedule [Klamroth and Weicek, p.20].

Solutions from Klamroth and Weicek show promising results. The solutions show the structure of the efficient and non-dominated set of the problem. The time-dependency shows the mutual relationships among the jobs of the efficient schedule, their order in the schedule with respect to time, and the related objective function values.

[Klamroth, *et al*. pp.24-25.] Using MATLAB, Klamroth and Weicek developed software to generate and organize the result. The resulting AMADEuS program is an interactive decision tool for data analysis and graphical output.

Another multiple criteria scheduling study on aircraft routing, crew pairing, and work assignment was done by Desrosiers, Lasry, McInnis, Solomon, and Soumis [Desrosiers, *et al*., pp.41-53]. In this research, they looked at a problem of planning and scheduling for an airline company. The goal was to streamline the planning process by optimizing aircraft routing, crew pairing, and work assignment. The airline company had purchased commercial airline operations management software called ALTITUDE [Desrosiers, *et al*., p. 42]. For the solution, Desrosiers formulated an optimization interface with ALTITUDE, that included routines and subroutines optimizing the three objectives. The resulting product generated results that are almost always near-optimal [Desrosiers, *et al*., p. 48].

In addition to a conventional MOLP formulation to arrive at an optimal solution, Steuer suggests that, in practice, interactive procedures have also proven to be most effective in de-conflicting criteria by searching the tradeoff space for a final solution [Steuer, p.4].   The interactive procedures involve a decision maker and machines to iteratively guide searches at each phase of a decision process.

Visual Interactive Modeling (VIM) Scheduling

A recent article by Belton and Elder [Belton and Elder, 1996, p.162] explores the iterative man-machine procedures introduced by Steuer.   Visual Interactive Modeling (VIM) was introduced by Belton and Elder as a way to explore solutions to a multi-

criteria production scheduling problem [Belton and Elder, 1996, p.162-174]. VIM uses

expert knowledge to guide the schedule generation process. It uses an interface to a

heuristic engine, with a built-in control mechanism to influence heuristic search,

preference, or performance criteria, to iteratively search the solution space.
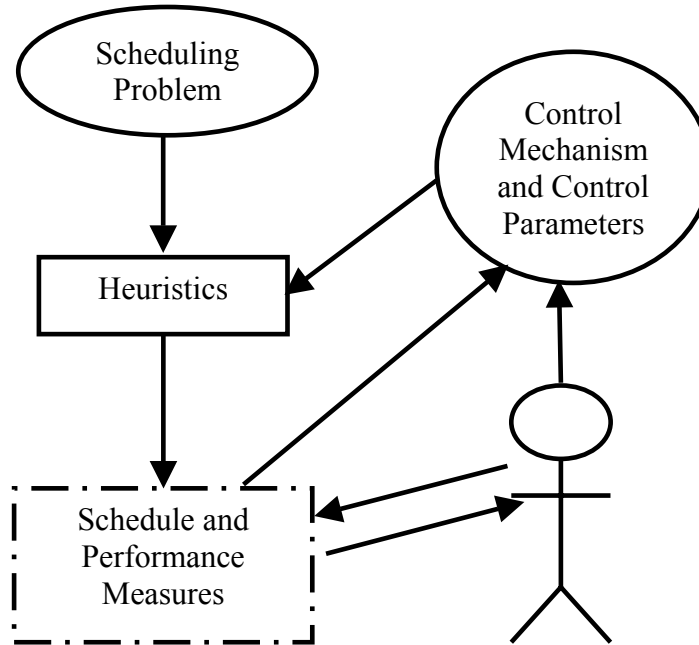
Figure 3. Belton and Elder's Visual Interactive Modeling

Evren also researched a method called Knowledge-Based modeling [Evren,

1999]. This Knowledge-Based approach is very similar to VIM in that both use expert-

knowledge to guide the scheduling process by generating and improving schedules until

an optimal schedule is obtained. With these methods, heuristics are employed to find a

good solution. From there, experts made modifications to either parameters (VIM)

[Belton and Elder, 1996] or schedules (Knowledge-Based) [Evren, 1999] to provide a

feasible and robust schedule. If software-based scheduling is being employed, VIM or

Knowledge-Based scheduling is a natural extension of the software. A diagram adopted from Belton's work (Figure 3) conceptually shows how VIM influences the scheduling process [Belton and Elder, 1996, p. 164].

The VIM concept springs from the disconnection of the input/output processes of scheduling [Belton and Elder, 1996]. Belton states that the scheduling heuristics being described in terms of criteria such as input data, job times, due dates, and other factors have no clear link to the output of the process where a schedule is produced only after the heuristics are applied. VIM provides the link by making the scheduling process interactive, with the scheduler using the control mechanism to iteratively change input parameters and guide directions to the heuristic search to produce new schedules [Belton and Elder, 1996].

In the Knowledge-Based scheduling approach, Noronha describes using algorithms or heuristics to obtain a baseline schedule. Once a baseline schedule is generated, a decision support system is employed by the expert to manipulate input parameters to further refine the schedule to meet criteria [Noronha, 1991]. The expert, or man-in-the-loop, controller ensures the improvement on the baseline schedule will generate a robust schedule.

Both Belton and Elder's and Evren's work show promising results in the use of VIM and Knowledge-Based scheduling. Belton and Elder's work show VIM is promising in generating a robust schedule. Belton and Elder also show that, given the setup of the software, sensitivity analysis can be performed to find the optimal schedule. They cautioned that sensitivity analysis on the input data sometimes did not show a clear

pattern. Belton and Elder emphasized that since they were using a simple weighted sum

as their priority rule to guide the search heuristic, improving the guide to the search

heuristic may yield improvement in the results. Evren states there were problems with

learning the new software and concept in the decision support system, but the software

and methodology shows promising results.

Programming Languages

To implement any heuristics or rule-based algorithms, programming languages

must be considered. To select the right programming language, considerations of the

language are based on the criteria of their: availability, ease to learn, ease of use-reuse,

and interoperability with existing software. A paper by Dupont, Nguyen, and Pektas

examined the three most common object oriented languages used today [Dupont, *et al*,

2002]. In addition to the language characteristics, they also looked at the environment

where the languages are best suited to be utilized. The three object-oriented

programming languages considered are C++, Java, and MS Visual Basic.

The most compelling reasons to use Visual Basic over Java and C++ are ease of

application integration, relatively quick learning time, and availability of host

environment [Dupont, *et al*, 2002]. The majority of desktop computers in offices and

homes today use a version of the Microsoft Windows operating system (Windows 95, 98,

2000, ME, XP or various versions of NT) [Kiely, Nov 1997, I656]. Since Microsoft also

develops the MS Office Suite on the foundation of the Visual Basic engine, they can

build enhancements and attachment modules into the application to solve specific

problems, and is assured a very high probability of error-free integration. Most people

are familiar with the Windows interface so the learning times to use the products are reduced [Kiely, Nov 1997, I656]. MS Office products such as Word and Excel have become the main word processor and spreadsheet in the majority of offices and homes. The fact that these products come already pre-installed when computers are produced certainly helps to increase familiarity with Microsoft Office products. The development environments and the required engine are already present in the Office products. The integrated development environment and ready-made templates for the user interface allows rapid development of any applications that use them. Once the applications are developed, the probability that the applications will work with its host applications is high [Kiely, Nov 1997, I656]. For these reasons, VBA was chosen for this project.

Current Scheduling Software Used at the 87th FTS

The squadron scheduler at the 87th FTS currently uses an MS Excel spreadsheet to generate the schedule. It is a large Excel workbook, with individual sheets for entering information, to generate the aircraft schedule, the academic classroom schedule, and the simulator-training schedule. Each schedule type has its own individual input and formatted output sheet. There are also the maintenance and simulator contracts generated for distribution to the respective shops.

Training Integrated Management System (TIMS)

The Training Integrated Management System (TIMS) is a new training management system currently being acquired by the Joint Primary Aircraft Training System (JPATS) System Program Office (SPO). TIMS is part of the JPATS Ground Based Training System (GBTS). JPATS GBTS is the complete ground portion of the

training environment and includes TIMS, computer hardware and other software, curriculum materials, and other resources. A detailed description of TIMS and the JPATS GBTS can be found in the Raytheon Aircraft Company's Software User's Manual for the Training Integrated Management System of the Joint Primary Aircraft Training System Ground Based Training System, hereafter will be referred to as TIMS User's Manual.

"The TIMS will manage Undergraduate Flying Training (UFT) for Air Education and Training Command (AETC) and Chief of Namal Educatin and Training (CNET)." [TIMS User's Manual, p.1] TIMS purpose is to integrate control and increase standardization to increase efficiency in the flying training processes across all the undergraduate pilot training programs.

TIMS is a large hardware- and software-based interconnected training management system. The hardware is a personal computer based client-server architecture. Clients and local servers are located at the different training bases. Master servers and databases are located at Randolph AFB. Local clients are connected to each other by local area networks. Wide area networks are used to connect the different training bases to each other and to Randolph AFB.

TIMS has many different components and functions to manage the flying training program. Included functions are: academic, administration, HQ administration, personal information, resources, training results, schedule building, schedule execution, and training syllabus tracking. These functions are replacements for the numerous separate components currently in use today. TIMS was designed to bring these functions together in one manageable environment.

This thesis concentrates on one aspect of TIMS, the schedule building function, that has the potential for adoption. TIMS's schedule building functions are currently very similar to the current process. Essentially, both the current schedule building functions and TIMS's process both build schedules manually. In the current process, the squadron scheduler builds the scheudles by manually entering the requests, then manually break the requests into the appropriate number of sorties per GO. Any changes are implemented by manually deleting the sortie and updating with the appropriate changes. Schedulers using TIMS will build schedules by manually selecting individual requests and required resources and placing them on a blank schedule. This thesis goes further by creating a model using scheduling rules to automatically generate schedules. Thus any model developed to support fighter pilot training should be flexible enough to either be able to interact with TIMS or be integrated into TIMS.

Summary

This chapter has presented an overview of the scheduling problem and the pertinent literature. With this material as a foundation, a methodology to address the 87th FTS scheduling process was developed. The next chapter covers the methodology to develop both the scheduling software algorithm and the analysis environment to test the software interface and the new scheduling algorithm.

Chapter III. Methodologies

Overview

      This chapter describes the methodology to be employed in the thesis. This chapter is partitioned into distinct areas: scheduling goals and objectives, the scheduling model, problem characteristics, scheduling rules, VIM, software design, software algorithm, notional training schedule, an attrition model, and the statistical analysis. The methodology used in the scheduling tool is also broken into several areas, each defining a critical component of the scheduling process: the scheduling tool to be developed for the 87th FTS, the algorithms being employed in the tool, and the statistics and analytical products generated from the scheduling tool. The first section gives the definitions of the objectives.

Scheduling Goals and Objectives

      The squadron scheduler produces the schedules and the simulator and aircraft contracts to meet certain goals and objectives. The schedules and contracts are utilized to ensure students receive adequate instruction to meet the training goals and timeline to graduate on time. In addition, students who fly more than the minimum required sorties have more opportunity to improve their flying skills. Thus a second goal is to maximize aircraft training time. Third, each aircraft requires maintenance preparation every day before any flying may take place. Any unused or underutilized aircraft is an inefficient use of manpower and other resources. A third goal is to minimize excess aircraft preparation and unused flights while still providing sufficient aircraft to assure that training objectives are met. The overall objective for the squadron scheduler is to

produce robust schedules that will satisfy these training goals. In addition, the objective

of the thesis is to provide a decision support scheduling tool that reduces time required to

build the schedules. Any amount of time saved by using the scheduling tool means that

much more time can be redirected to training the students. Figure 4 gives an overview of

the squadron scheduling inputs and outputs.
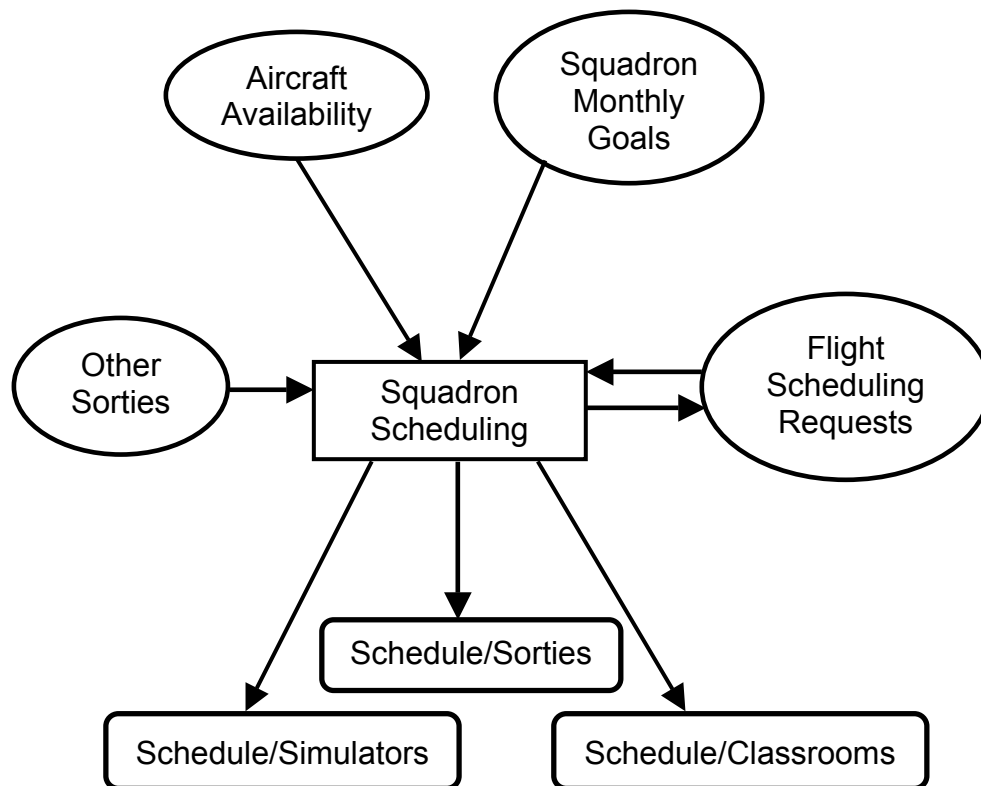
Figure 4. Squadron Scheduling Products

The Scheduling Model

To begin the scheduling model, first look at the scheduling process at a high-level

view. The detailed scheduling process can be found in the 87th FTS scheduling

procedure manual (87th FTS Scheduling Manual, p. 1). The scheduling process can be summarized by six steps:

1. The four flights submit their flight schedules for review.

2. The squadron scheduler considers data from: maintenance aircraft availability, monthly squadron goals, and other sorties requests.

3. Squadron scheduler generates a draft schedule.

4. Squadron scheduler confirms the draft schedule with the flights. If there are scheduling conflicts, the squadron scheduler de-conflicts the problem by reassigning requests or resources in the schedules. Repeat step 3.

5. The finished draft schedules are submitted to the maintenance squadron. If there is a conflict, go to step 4.

6. Squadron scheduler and maintenance accepts the schedule and contracts aircraft for the scheduled week. The squadron scheduler and the simulator shop accept the simulator contract for the scheduled week.

Problem Characteristic

The scheduling environment must be understood before scheduling can be performed. The scheduling environment can be understood in terms of its dynamic changes, the schedules' requirements, and other scheduling constraints.

The training environment at the 87th FTS is a fluid and dynamic environment characterized by daily, changing requirements. Operational, maintenance, and weather cancellations can happen daily. Requirements and priority changes to schedules occur frequently. In addition to the changes in schedules, the squadron scheduler also changes periodically. The different types of cancellations and other scheduling requirements initiate a *re-scheduling* of the schedules.

Different types of cancellations can stress the squadron schedules. Operational cancellations can be attributed to the students, a scheduling issue, or other issues. Students often "bust a ride" during aircraft sortie training. A student might become sick during a ride and be unable to complete the sortie training. This is counted as an operational cancellation. Less proficient students may simply fail the maneuvers or do not meet the minimum grade for the aircraft sortie training session, also causing an operational cancellation. A plane might be rejected because of a pre-flight inspection or malfunction before or during sortie training, requiring the aircraft sortie training session to be aborted. If there have been a number of grounded aircraft or longer than expected cycles, there may be no planes available. These are counted as maintenance cancellations.

Weather plays an important factor in aircraft sortie training. The weather might be too foggy or cloudy to meet visibility minimums for student pilots. A thunderstorm or severe heat will ground all student pilots from taking off until the weather improves. Other weather patterns can also cancel aircraft sortie training.

Other requirements may also affect the schedules. A general officer or other VIP visiting the base requests a sortie ride. Students fall behind the syllabus timeline and need additional training to catch up with the class. They require extra aircraft sortie or priority scheduling outside the normal flight priority. A number of special requirements can occur that force the aircraft sorties to be re-scheduled.

The squadron scheduler also frequently changes. Squadron scheduling is an additional duty for an experienced instructor pilot. As such, the instructor pilots are rotated at approximately every six months to preserve the instructor pilot's flying ability

and to provide new IPs an opportunity in the scheduling shop. Often, the new squadron scheduler possesses little to no squadron scheduling experience. An inexperienced squadron scheduler may spend the first two months learning the scheduling process. Once the squadron scheduler is proficient in the scheduling process, it is often time to rotate the position of scheduler to an inexperienced replacement. The result is a loss of experience from the turnover of the schedulers. While the previous scheduler may be available for consultation, ultimately the new scheduler must "solo" on squadron scheduling.

The squadron scheduling shop produces the training, simulator, and classroom schedules. The three schedules generated have different priorities and other requirements.

The aircraft sortie schedule has several unique requirements. The aircraft schedule assigns runway take-off times to student-instructor pairs supplied by the flights. The aircraft sortie schedule must also consider training times for student aircraft controllers during certain periods. Thus only senior students should be assigned sorties during the training time slots for student aircraft controllers and they should not be practicing advanced maneuvers.

The aircraft sortie schedule is broken up into three different take-off periods or "GOs". The GOs are determined by the turn-time of the aircraft. A typical aircraft has a 1 hour 20 minutes mission time and 1 hours 20 minutes of maintenance turn-time. Thus a typical aircraft can fly a sortie and be ready to fly again in 2 hours 40 minutes. Allowing time for student/IP aircraft check and acceptance time, and waiting time for the runway, and the typical time extends to approximately 3 hours 10 minutes. Therefore 3

hours 10 minutes is the typical length of a GO. A typical given daylight window consists of a maximum of three GOs.

One goal of the training program is to provide the students with the maximum available training in the aircraft sortie. Thus the aircraft sortie schedule receives the highest priority when schedules conflict. The simulator and classroom schedules work around the aircraft sortie schedule when possible.

The simulator schedule provides training schedules for simulator training. There are four simulator pods available. Thus, only four students are able to fly on simulator sorties at any given time. In a typical day, a maximum of 20 to 24 sorties can be generated. The simulator pods are a limited resource and must be contracted with the simulator shop. A simulator sortie requires an experienced IP to accompany the student. This typically consists of contractor IPs with prior service experience. A simulator contract includes the required number of simulator sorties and the accompanying IPs. Any extra sorties outside the contract will incur additional costs. Simulator schedules are of lower priority than aircraft sortie schedules, but are of higher priority than classroom schedules.

The classroom schedule assigns available classrooms for different instruction. Computer aided instructions (CAI) are independent study sessions, while instructor based training (IBT) requires one or more experienced and qualified instructors. IBTs are taught to the whole class within a flight, thus requiring all students in the same class to be available for training. Those students cannot be assigned to aircraft sortie or simulator sortie training during this time.

Classroom instructions are usually scheduled late during the day. This allows students to fly sorties during the calm morning and early noon weather (subject to seasonal weather variations.) If a morning sortie requires re-scheduling on a spare aircraft, the schedule allows the re-scheduling between the first and second, or second and third GO without interfering with the classroom schedule. Daylight hours are shorter during the winter, thus scheduling classes in the late afternoon and into the evening allows maximum utilization of daylight hours for aircraft sortie training. Simply put, classroom scheduling provides the most flexibility to training times and are placed at the end of the day to take advantage of this.

There are several constraints associated with the aircraft sortie schedule. Most aircraft sorties are limited to the daylight hours (the exception is night flying training.) The time slots for aircraft take-offs in the window of daylight hours, from sunup to sundown, must be shared between the eight different classes. A typical aircraft sortie turnaround time is 3 hours 10 minutes. Other time requirements are students sortie training turnaround times: pre-brief, training, and post-brief times. Pre-brief usually lasts 45 minutes to 1 hour. Aircraft sortie training typically lasts 1 hour 20 minutes. Post-brief activities last 45 minutes to 1 hour 10 minutes. The AETC Training Syllabus requires a minimum turnaround time of 2 hours 45 minutes [AETC Syllabus, p.3]. A typical student sortie turnaround time is 3 hours 30 minutes, with a minimum of 3 hours 10 minutes.

Simulator training requires similar requirements to aircraft sortie training. Simulator briefs are typically the same as aircraft sortie training. Pre-brief, training, and post-brief times are similar, with 3 hours 30 minutes as a typical turn-time. Since the simulators are housed indoors, simulator training is not constrained to daylight hours.

However, simulator training usually occurs during the aircraft sortie training due to the instructor-based training requirement.

Instructor based training (IBT) occurs inside a classroom. IBT is taught by one or more experienced instructors and is taught to the full class body. The full class is taught at one time to limit duplication of efforts by the instructors.

Other constraints that can affect the schedules are categorized as aircraft and training constraints. After an aircraft sortie schedule is produced, the number of aircraft required to meet the training is contracted to the maintenance squadron. Any additional aircraft above the contracted number of aircraft costs the squadron additional funds. In addition to the required number of aircraft, there are typically eight aircraft contracted as spares. These aircraft are used to replace aircraft down for different reasons and to allow students to re-fly sorties aborted for other reasons. The spare aircraft are also used to provide additional training to students.

Some training limiting factors constrain the schedules produced. A student must stand down for 12 hours of crew rests after each day's training. Students are also limited to three training events, not including academic training. If three flying sorties training occur in a day (flying sorties training includes both aircraft and simulator sorties,) there can be no more than two hooded aircraft sorties, two simulator sorties, or a combination of two of one type of sorties and the third another type [AETC Syllabus, p.3]. A hooded aircraft sortie is flown with an opaque hood pulled over the canopy of the aircraft to simulate severe weather. This simulates a severe environment where the student is flying blind, using only instruments to guide the aircraft. A more detailed description of

training and other constraints can be found in the AETC training syllabus

[AETC Syllabus, pg.1-7].

In a training environment, any of a variety of factors can affect the schedules. The example cancellations and factors discussed previously can affect the static schedule. The dynamic changes often require minute readjustments to the original schedule to keep the goals and objectives satisfied. Sometimes, the multiple goals and objectives can conflict with each other. Recent research by Belton and Elder shows VIM can be used to interactively de-conflict the goals and objectives to create feasible schedules in a reasonable time.

Visual Interactive Modeling

Belton and Elder introduced Visual Interactive Modeling in a 1996 article [Belton and Elder, 1996] discussed in Chapter II. Belton and Elder's VIM concept de-conflicts multiple conflicting goals by exploring the solution space to find an "acceptable" solution. The VIM is a framework where the expert interacts with the tool to guide the search in the solution space to find an acceptable schedule. In this thesis, the VIM concept is utilized in conjunction with the software interface and the scheduling rules built within the software to find a robust schedule. The scheduling rule generates an initial schedule. The squadron scheduler makes iterative adjustments to the initial schedule until an acceptable schedule is found.

Within the VIM framework, the squadron scheduler interacts with the software throughout the scheduling creation process. The squadron scheduler generates an initial schedule by choosing the desired scheduling rules based on his judgement and the current

operational environment. The squadron scheduler checks the generated schedule to make

sure it satisfies all requirements. He confirms the schedules with the flight schedulers,

the aircraft maintenance shop, and the simulator shop. If changes are needed, the

squadron scheduler generates subsequent schedules by modifying the initial schedule.

The process repeats until a robust schedule is produced. Figure 5 shows the VIM

interaction in the scheduling process.
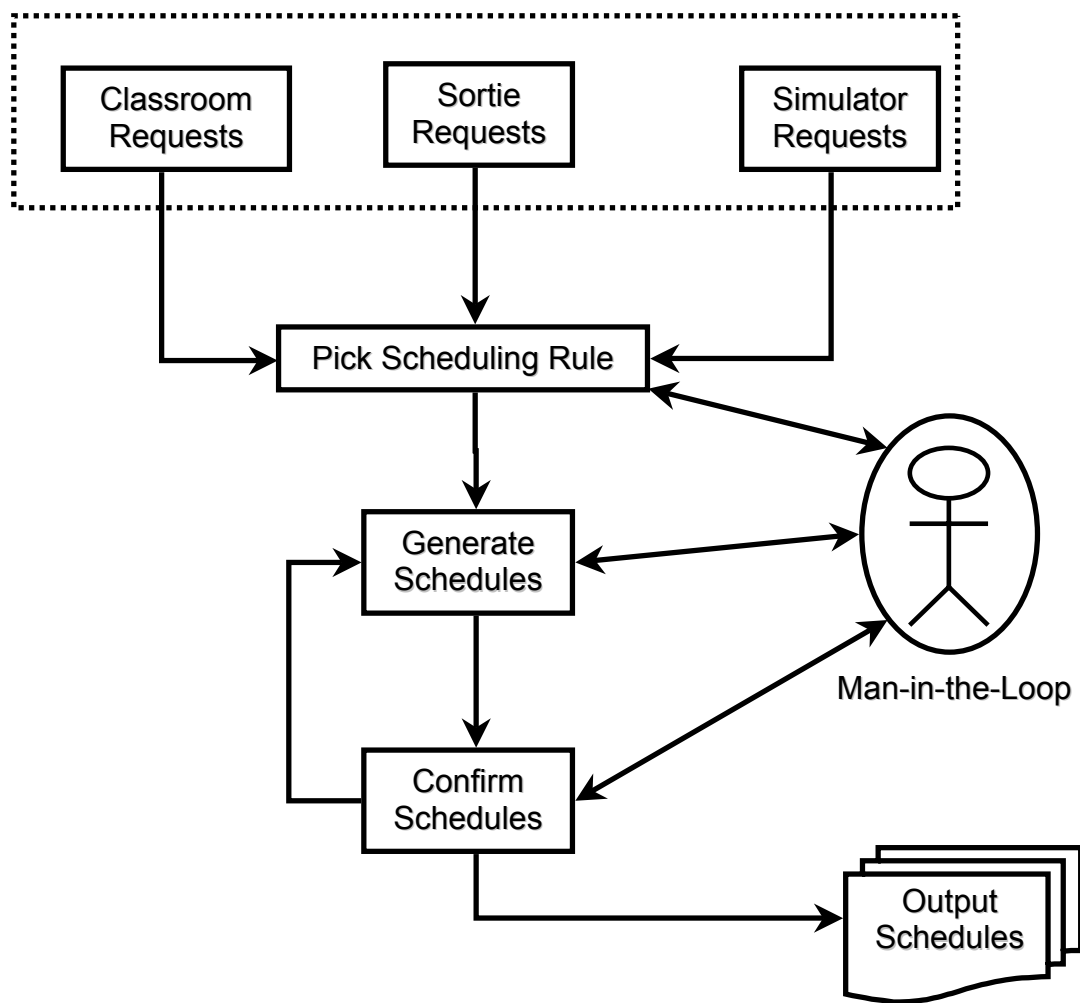
Figure 5. VIM Interaction in the Scheduling Process

Scheduling Dispatch Rules

After inputting the required requests, the second stepin the interactive process is choosing the appropriate scheduling dispatch rules. Dispatch rules are used to find initial solutions for several reasons. Belton and Elder's VIM uses an expert to guide the heuristics to search for a good schedule in a short time. In the squadron scheduling environment, requirements and priorities change daily. The schedules generated in the squadron scheduling shop are deterministic schedules, generated and finalized one week in advance. The advance scheduling is required because the resources required for the execution of the aircraft sortie and simulator schedules have to be contracted with the respective maintenance and simulator shops. Because of the dynamic changes that occur in a training environment, once the schedule is finalized, optimized schedules produced the week prior are often no longer optimized once the dynamics of daily changes occur. The schedules that best meet the squadron's needs must be flexible and robust and be able to allow the changes to occur without significantly changing the original schedules. Pinedo shows some general purpose procedures used for deterministic scheduling that produces reasonably good solutions in a relatively short time [Pinedo, 1995, p.142]. Using the general dispatch rules, the squadron scheduler can quickly create initial feasible schedules and, through the VIM process, iteratively modify the initial schedule to arrive at schedules that satisfy the scheduler's requirements.

By investigating the possible types of priorities the squadron scheduler might use to prioritize the flights, appropriate scheduling dispatch rules can be chosen for the model. With daily dynamic changes, the squadron scheduler might prioritizes the flights' requests according to the current specific needs. A flight with the largest number of

requests for one day might receive the highest priority to be first on the schedule in order

to be flexible in case a student needs to re-schedule due to cancellations. On another day,

a flight that is behind in training schedule requirements might receive first priority in the

schedule to catch up. The three scheduling priorities used for this thesis are: Largest

Number of Requests, Flight Behind the Training Schedule the Most, and Class Seniority.

These priorities correspond to the scheduling dispatch rules: Longest Processing Time,

Minimum Slack, and Least Flexible Job, respectively.

Table 1. Scheduling Priorities and Dispatch Rules

Squadron Scheduling Priority	Scheduling Dispatch Rules
Largest Number of Requests	Longest Processing Time (LPT)
Flight Behind Training Schedule	Minimum Slack (MS)
Class Seniority	Least Flexible Job (LFJ)

Software Design and Implementation

Once the scheduler chooses the appropriate scheduling dispatch rule, the

scheduler interacts with the decision support tool through the software interface to make

iterative adjustments of the schedule. The software is designed around the existing Excel

spreadsheet and utilizes the existing programming ability inherent in Excel to provide an

improved interface. The software interface is designed around three areas: familiarity,

flexibility, and user friendly.

Software design and implementation takes advantage of existing tools/software,

as covered in Chapter II, to speed up the creation process. The existing tool was created

in Excel. Inherent in Excel is the Visual Basic for Applications (VBA) programming

language. VBA can thus be used to extend the existing tool by programming in

additional capabilities using VBA codes. Since VBA is a native programming language to MS Excel, VBA is seamlessly integrated into the tool and requires no new software. Any VBA coding in Excel can be seen, modified, used, and re-used by other programmers. Any new functions can be easily added to the existing tool.

The existing MS Excel scheduling tool has been used to create past schedules and is familiar to the squadron's schedulers and commanders. The existing Excel scheduling tool already has a defined scheduling output format. By using the existing format, the squadron scheduler's familiarity with the scheduling layout assists the scheduler in the scheduling process. The existing spreadsheet already possesses the required official format to the different outputs for the schedules and contracts. Past squadron schedulers have written instructions on what information needs to be entered and what information needs to be updated to create the various schedules in the existing format. Finally, past squadron schedulers and commanders are familiar with the current products. Using the current Excel spreadsheet will not require any additional training to become familiar with the enhanced decision support tool.

A key improvement to the existing Excel spreadsheet is flexibility in both entering and manipulating data. Existing spreadsheets have built-in formulae in the existing cells where the sortie schedule is displayed. Any accidental deletion in the cells would have destroyed the formula in the cell. An inexperienced Excel user might not know how to retrieve or replace the required formula, essentially destroying the scheduling tool making it useless until a competent person can be found to fix the problem. A typical sortie schedule is shown in Table 2.

Table 2. A Typical Sortie Schedule

Tuesday

T/O	FLT	T/O	FLT	T/O	FLT	T/O	FLT
0730	N	1040	N	1350	O	0000	M
0735	N	1045	N	1355	O	0005	M
0740	N	1050	N	1400	M	0010	M
0745	N	1055	M	1405	M	0015	M
0750	N	1100	M	1410	M	0020	M
0755	N	1105	M	1415	M	0025	M
0800	N	1110	M	1420	M	0030	M
0805	L	1115	M	1425	M	0035	M
0810	L	1120	M	1430	M	0040	M
0815	L	1125	M	1435	M	0045	M
0820	L	1130	M	1440	M	0050	
0825	L	1135	M	1445	L	0055	
0830	L	1140	L	1450	L	0100	
0835	L	1145	L	1455	L	0105	
0840	L	1150	L	1500	L	0110	
0845	L	1155	L	1505	L	0115	
0850	L	1200	L	1510	L	0120	
0855	O	1205	L	1515	L	0125	
0900	O	1210	L	1520	L	0130	
0905	O	1215	L	1525	O	0135	
0910	O	1220	L	1530	O	0140	
0915	O	1225	L	1535	O	0145	
0920	O	1230	O	1540	O	0150	
0925	N	1235	O	1545		0155	
0930	N	1240	O	1550		0200	
0935	N	1245	O	1555		0205	
0940	N	1250		1600		0210	
0945		1255		1605		0215	
0950		1300		1610		0220	
0955		1305		1615		0225	
1000		1310		1620		0230	
1005		1315		1625		0235	
1010		1320		1630		0240	
1015		1325		1635		0245	
1020		1330		1640		0250	
1025		1335		1645		0255	
1030		1340		1650		0300	
1035		1345		1655		0305	
SORTIES	27		26		23		10
TOTAL	27		53		**76**		**86**
				CAP	27		

38

Another key flexibility point is the ability for the scheduler to manipulate scheduling data to create schedules. The scheduler is able to directly enter scheduling data into a schedule's cell. The scheduling algorithm recognizes there is a hard requirement in the schedule and will build the remaining schedule around the hard-coding data. This gives the scheduler the ability to quickly enter hard requirements and build other requirements around it without having to do very much manipulation of the software.

Finally, the decision support tool is populated with buttons and menus to present a user-friendly interface. An opening menu gives a list of options from creation of schedules, to reviewing schedules, to printing out various schedules and contracts. Buttons are placed in various locations throughout the spreadsheet and provide the scheduler with options to clear the current schedule template to generate new schedules for either one specific day or the entire week. Buttons, representing dispatch rules or flight priorities, provide the scheduler with a direct link to the scheduling rules to generate schedules for one day or the whole week. Other buttons update the scheduler's preference in flights priorities.

Scheduling Algorithm

Behind the flight priorities buttons are the corresponding scheduling dispatch rule algorithms. The scheduler pushes the flight priorities buttons to activate the scheduling algorithm to generate the initial weekly schedule. Using the VIM concept, the scheduler interacts and adjusts the daily schedule by changing specific requirements for that one day. Re-scheduling the specific day requires pushing the appropriate day's re-scheduling

button. The scheduler repeats the process until all five days are complete, if desired.

Table 3 shows the steps to the scheduling. Figure 6 shows the VIM flowchart using the

scheduling algorithm.

Table 3. The Scheduling Algorithm

<div style="border:1px solid black">

The Scheduling Algorithm

1. Scheduler input flights' requests and special requirements.
2. Scheduler selects appropriate flight priority (dispatch rule)
3. Software prioritize flight requests based on selected priority (dispatch rule)
4. Assigns flight request based on the priority to appropriate time-slots
   a. Assign sortie requests
   b. Assign simulator requests
5. Scheduler modify schedule if necessary. Repeat step 2 if new prioritization is needed.
6. Finalize schedules.

</div>

Notional Training Schedule

While the description of the Squadron Scheduling Decision Support Scheduling

tool has been completed, an experiment was developed to debug and test the tool. This

section described the simulated environment developed to test the *SSDST*.
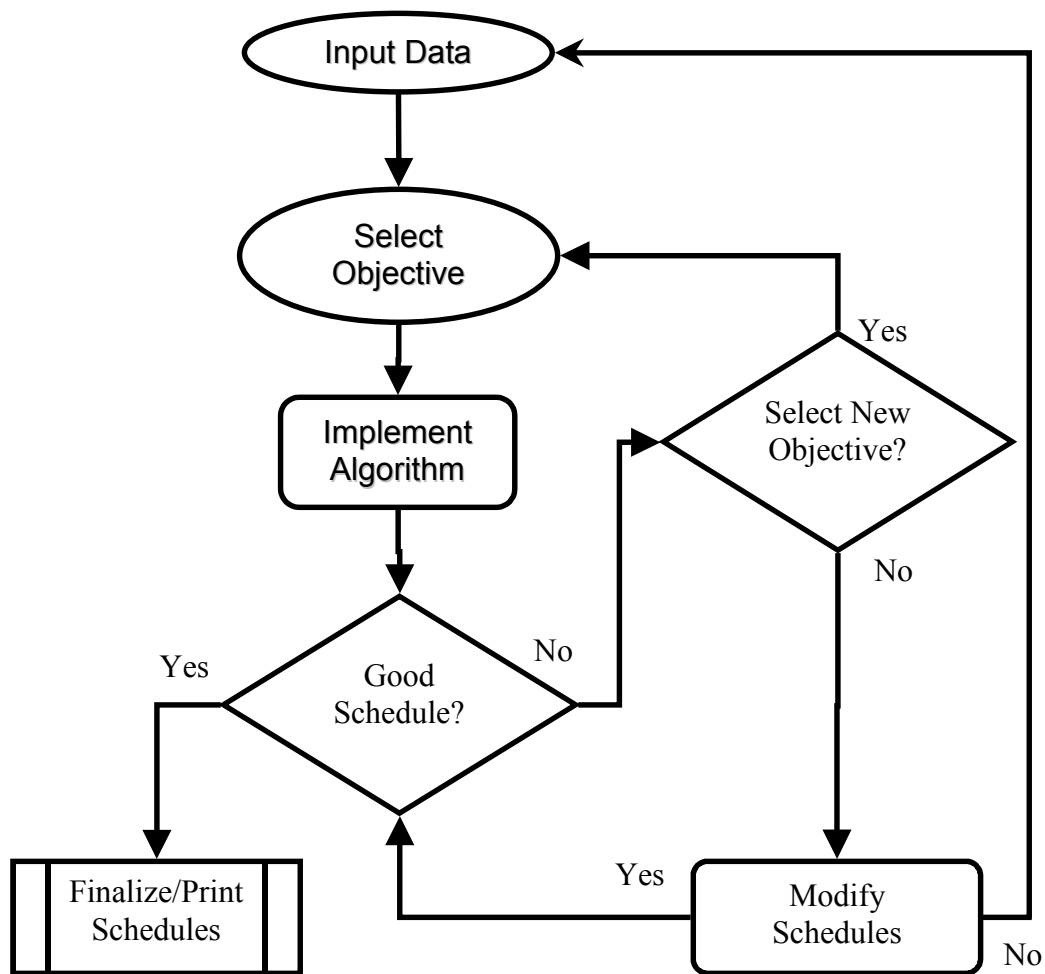
Figure 6. Software Implementation Using VIM

A notional training schedule was developed to test the scheduling dispatch rules

and its implementation in the software. The notional training schedule is based on the

actual training requirements taken from the AETC Training Syllabus [AETC Syllabus,

pp. 1-62]. The notional training schedule is based on the 120-day training program

duration. Since the squadron schedule is a weekly schedule, the notional training

schedule requirements are laid out in terms of weekly requirements. The weekly

requirements can then be divided into daily requirements during the scheduling process.

The notional training schedule simulates a steady training environment. The AETC training syllabus provides a complete list of training requirements in terms of aircraft sorties, simulator sorties, and academic and other ground training. From the list an evenly distributed training schedule is developed. The logical reason for a steady schedule is to provide the students with a steady training flow to promote maximum learning. The notional training schedule is a 5-days schedule and does not include weekend cross-country flights. Table 4 and Figure 7 shows the notional training schedule for this thesis.

Table 4. 24 Weeks Notional Sortie Training Schedule

Training Week	1	2	3	4	5	6	7	8	9	10	11	12
Training Requirement	0	2	2	3	3	5	5	5	5	5	5	5

Training Week	13	14	15	16	17	18	19	20	21	22	23	24	Total
Training Requirement	5	5	5	5	5	5	5	5	5	4	3	3	100
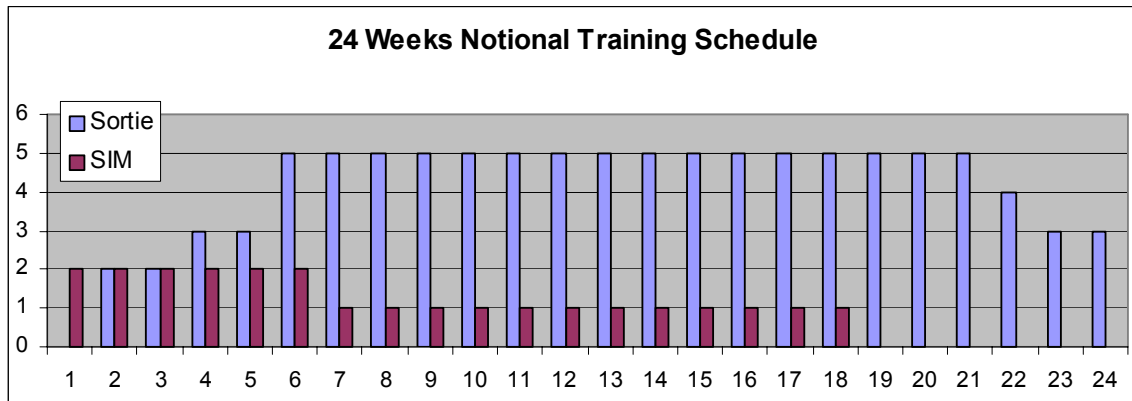
Figure 7. 24 Weeks Notional Training Schedule

42

The notional training schedule also simulates the dynamics of the training classes. Approximately every three weeks, a new class begins training and a senior class graduates. There are four flights at the 87th FTS, thus a flight consists of two different classes at any period of time. The training schedule differentiates between the different classes to better captures the realistic training environment. The training program at Laughlin is opened year round. Thus a snapshot of the 120-day training program consists of flights at various stages of the training program. To simulate a snapshot of the training program, the simulator creates eight different flights and their corresponding training requirements.

Table 5. Classes in Their Respective Training Program Status

Class	Num Students	Week Status
L1	11	1-3
L2	11	4-6
M1	10	7-9
M2	10	10-12
N1	9	13-15
N2	9	16-18
O1	8	19-21
O2	8	22-24

Attrition Model

An attrition model was created to simulate the different attrition types and percentages. The attrition model is used to simulate the attrition of sorties that can typically be found in a training environment. The attrition model includes attrition for operational, weather, maintenance, and other attritions. A historical rate or percentage is used for the model. The squadron scheduler and the maintenance shop have used the

historical attrition rate to plan for sortie requirements. The thesis author tried to verify

the source of the attrition rate, but was not able to pin point its origin. For this thesis

analysis, the assumption is the historical rate is valid.

Table 6. Historical Attrition Rates

Attrition	Historical
OPS	0.00%
WX	13.40%
MX	0.40%
OTHER	2.10%
TOTAL	15.90%

The attrition rate also affects the weekly sortie requirements. Using the 15.9%

attrition rate, the squadron scheduler planned for sortie attrition by adding additional

sorties to the base sortie rates. Simulator sorties are not affected by aircraft maintenance

or weather attritions, thus the rates are unchanged.

Once the schedule is generated, a post-scheduling attrition model is applied to the

schedule during statistical analysis. The post-scheduling attrition model is applied to

each sortie scheduling event and any re-scheduling event to determine whether it is

affected by attrition. If the sortie scheduling or re-scheduling event is affected, the post-

scheduling attrition model determines which type of attrition it is. The result is passed

back to training schedule and shows up as either a "good" event or as a type of failure.

Failure types that are based on factors outside of the student's ability, such as

maintenance and weather, may be re-scheduled. Re-scheduled sorties are affected by

attrition at the same rate as the original sortie schedules. The attrition types are assigning

the attrition a percentage of the total attrition. The attrition types are assigned a section

of the percentage of the total attrition rate. A uniform distribution is used to draw

random numbers to determine whether the sortie scheduling or re-scheduling event is

affected or not.

**Table 7. Sortie Requirements Adjusted for
Attrition and Unadjusted SIM Requirements**

Adjusted Sortie Requirements					Unadjusted SIM Requirements				
	Students					Students			
Weeks	11	10	9	8	Weeks	11	10	9	8
1	0	0	0	0	1	22	20	18	16
2	27	24	22	20	2	22	20	18	16
3	27	24	22	20	3	22	20	18	16
4	40	36	33	29	4	22	20	18	16
5	40	36	33	29	5	22	20	18	16
6	66	60	54	48	6	22	20	18	16
7	66	60	54	48	7	11	10	9	8
8	66	60	54	48	8	11	10	9	8
9	66	60	54	48	9	11	10	9	8
10	66	60	54	48	10	11	10	9	8
11	66	60	54	48	11	11	10	9	8
12	66	60	54	48	12	11	10	9	8
13	66	60	54	48	13	11	10	9	8
14	66	60	54	48	14	11	10	9	8
15	66	60	54	48	15	11	10	9	8
16	66	60	54	48	16	11	10	9	8
17	66	60	54	48	17	11	10	9	8
18	66	60	54	48	18	11	10	9	8
19	66	60	54	48	19	0	0	0	0
20	66	60	54	48	20	0	0	0	0
21	66	60	54	48	21	0	0	0	0
22	53	48	43	39	22	0	0	0	0
23	40	36	33	29	23	0	0	0	0
24	40	36	33	29	24	0	0	0	0
Total	1323	1200	1083	963	**Total**	264	240	216	192

Table 7. Cumulative Attrition Assignments

Attrition	Cumulative
OPS	**0% - 0%**
MX	**0% - 2.5%**
WX	**2.5% - 86.8%**
OTHER	**86.8% - 100%**
TOTAL	**100.00%**
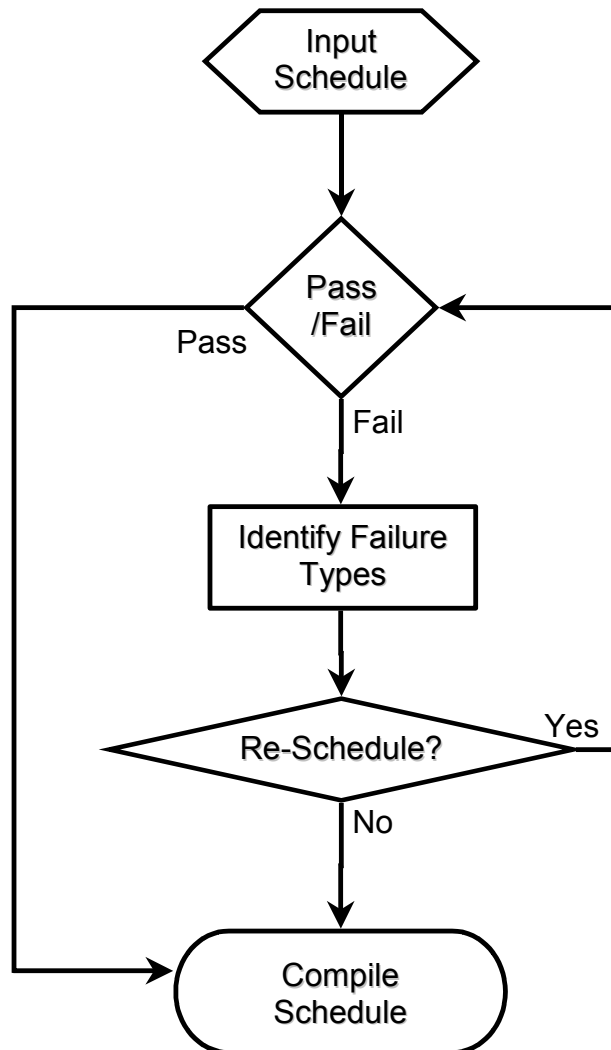
Figure 8. Post-Scheduling Attrition Model

Post-Scheduling Attrition Model Algorithm

1. Input schedule for analysis.

2. Draw random number from uniform distribution to determine attrition. If sortie failed, go to 3. Else, go to 5.

3. Determine attrition type.
4. Does the sortie need to be re-scheduled? If yes, re-schedule the sortie and go to 2. Else, go to 5.

5. Compile schedule with identified attritions.

Statistical Analysis

Statistical analysis reveals the performance of the decision support tool. There are several statistics of interest.

Questions being asked on the performance of the decision support tool are: is it fast? Flexible? And accurate?

Question on the dispatch rule: does it produce feasible schedules?

Question on the overall training goal: does it produce schedules that meet the students' training needs?

It should be noted that the simulated envoronment created was **NOT** intended to be a simulation of the entire training process. It is merely intended to pilot test the squadron scheduling decision support tool developed. A complete simulation of the training operational environment would be a major study in its own right. While the simulation is available to the 87th FTS, it is assumed that their intent will be focused in the squadron scheduling decision support tool.

Chapter IV presents the analysis undertaking in excercising the the scheduling tool in the simulated training environment.

Chapter IV. Analysis and Results

General

This chapter covers the analysis of the schedules generated to meet the simulated environment. This chapter is broken up into several sections. The first section sets up a notional training program, with varying flights consisting of different numbers of students. The second section analyzes the physical structure and performance of the software. The third section analyzes and summarizes the statistical analysis of the notional training program. The fourth section analyzes the different scheduling dispatch rules to determine effectiveness in generating feasible schedules that meet the students' graduation needs.

Notional Schedule Setup

The notional training program is created to test the software for analysis. The notional training program ideally should represent a real system by including as many of the features and characteristics of the real system as possible. The notional training program has eight different classes at different stages of training. Two classes are assigned to a flight, thus the squadron has four different flights. The classes in each flight have different numbers of students. The daily training requirements for each flight, based on the training syllabus, are detailed in Appendix B. In addition to the regular training, the notional schedule also includes night flying training at the 20th week and cross-country flights at the last three weekends of the training program, on the 22nd, 23rd, and 24th weeks of training.

Table 8. Training Schedule Starting Timeline

Classes	L1	L2	M1	M2	N1	N2	O1	O2
Number of Students	11	11	10	10	9	9	8	8
Starting Week	1	4	7	10	13	16	19	22

Eight different classes and characteristics are created for the notional

environment. The eight classes are representative of those shown in Table 7, Chapter III.

The eight classes are in various stages of training and are separated from each other by

about three weeks of training. The daily training requirements change as the classes

progress through the training program. As senior classes exit the training program, new

classes enter the program to replace the old. The classes also have different number of

students in each class. Daily requests are updated according to the status of the classes in

the training program.

Physical and software performance

Once the weekly requests are entered, the software takes over and builds an initial

schedule. The squadron scheduler makes necessary adjustments to the schedule until a

final solution is reached. During this process, the software design and software

performance can be measured.

The software design can be measured by looking at the interface environment and

the flexibility of the software in the schedule generation process. The software interface

environment is straightforward and user friendly. The introduction screen provides two

buttons, giving the option to go to the main menu to select choices or to exit the program

completely. The main menu provides a list of choices for the user to enter data into formatted and tabbed forms, review the schedule, print out the various scheduling product, exit directly to the spreadsheet, or exit the program completely.

The Excel spreadsheet is populated with various buttons, each with its own function. Some buttons assist in preparing the schedules by clearing data fields where sorties data are entered. This ensures there are no extraneous data being erroneously included into the schedule. Other buttons activate the scheduling algorithms. These buttons have VBA codes written in the background. Pushing an appropriate button activates the codes for the particular dispatch rule. The buttons are large and easy to recognize. If the buttons are too obtrusive, there is also an option to hide the buttons for better view of the draft schedule.

The Excel spreadsheet allows manual overrides of most functions to provide the squadron scheduler with maximum flexibility. The squadron scheduler may use one of the priority rules, or set his own flight priorities. A button allows the scheduler to update the entire week's priorities with one push of a button. The spreadsheet also allows manual override of the scheduling process. The scheduler may directly enter a request at a specific time of the day, and the scheduling algorithm will build the available schedule around this request. To represent blacked-out time slots, the scheduler puts holding spaces where a request normally resides, and the cells will automatically change color to highlight the cells as reserved empty or "blanked out".

Once the data is entered and the appropriate priority button is pushed, the scheduling algorithms build a draft schedule. The scheduling algorithm is fast in generating the initial schedule. Running on a 700 MHz computer with 128 MB RAM,

the algorithm generates the sorties and SIM weekly schedules in less than 16 seconds. Based on the squadron scheduler's current computer, an 800 MHz desktop with 256 MB RAM, the algorithm would be expected to generate schedules as fast as, or faster than, the test machine. Adjusting each individual day's scheduling requires additional time, depending on how much adjustment is needed. Confirming the actual number of aircrafts available and the flights' training requirements also take additional time, depending on the required information. In all, the scheduling algorithm provides an initial, usable schedule in a reasonable amount of time.

Statistical Analysis and Performance of the Scheduling Dispatch Rules

The notional training environment was used as training input to generate schedules for the analysis. The setup includes eight different classes combined into four flights. The classes are at different stages of the training program and are approximately three weeks apart. The three dispatch rules being measured for the analysis are the Least Flexible Job (LFJ), Longest Processing Time (LPT), and Minimum Slack (MS) corresponding to the flight scheduling priorities of Class Seniority First, Largest Number of Requests First, and Flight Furthest Behind the Training Schedule First, respectively.

The notional daily training schedule, as shown in Appendix B, is used as input for the schedules. Each week's requests are entered, and each scheduling rule is used to generate a schedule. The schedule is recorded for analysis. The weekly scheduling is repeated, with the classes' training status updated when necessary. After the 120-day training interval is completed, the data is analyzed.

After the schedule is generaged, the post-scheduling attrition model (Figure 8, Chapter III) is applied to each individual sortie to determine whether it is a good sortie, or if ineffective, then which type of cancellation it is. Chapter III mentioned that certain sortie may be re-scheduled for later during the day. The sortie being re-scheduled uses one of the eight spare aircraft for the day. If there are fewer sorties needing re-scheduling, it is assumed the rest of the aircrafts are used up by the priority class as additional training sorties in order to improve the priority class's training and to minimize unused aircraft at the end of the day. The process continues, and weekly statistics are compiled until the complete 120-day training schedule is complete. The process is applied to all three scheduling dispatch rules.

Requested Sorties for the Notional Training Schedule

The first statistic compiled is the *requested sorties*. Figure 9 shows the weekly requested sorties. The requested sorties data are ordered with the first week of the junior class as week 1. It is prudent to make sure that the sorties requests are correct. If the number of sorties requested are wrong, it will throw off the remaining output statistics. As expected, the sorties requested are very consistent between the four flights. The sorties requested per student are the same; the difference in the numbers are due to the difference in the number of students in each flight. The variations at the beginning and end of the training schedule are expected. The flights are ramping up their sorties training at the beginning so we observe a steady increase. Toward the end of the training schedule, again note a huge increase-decrease-increase in the number of sorties. The significant increase at week 19 is due to the junior class in the flight still requiring a

maximum number of sorties in the week while the senior class just enters their cross-country training weekend. The cross-country training occurs Friday, Saturday, and Sunday of the training week, thus adding an additional three sorties per student for that week. The sharp decrease from week 21 to week 22 is due to the senior class in the flight graduating so the new class entering the training program has no sortie training for the first week. Now that face validity for the sorties requests for the notional training schedule has been established, the scheduling heuristics are examined.
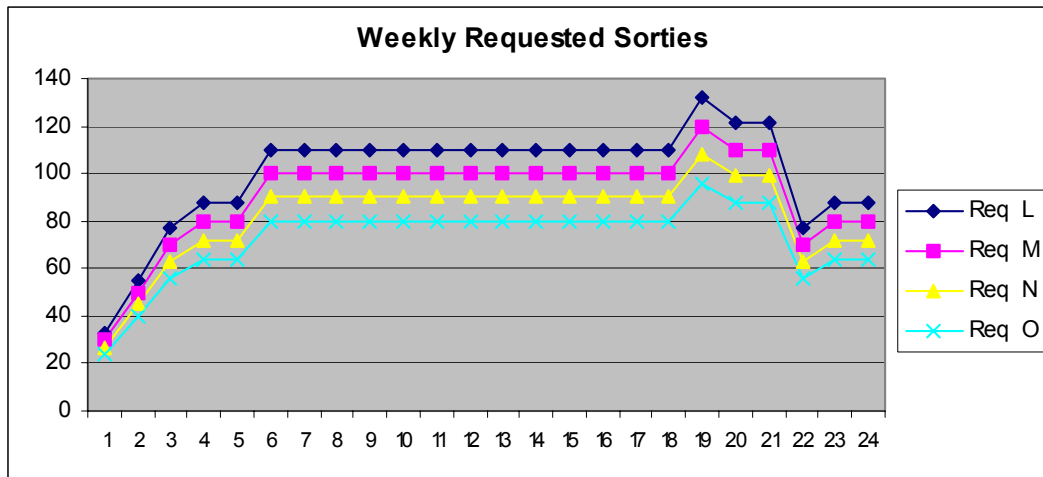
Figure 9. Weekly Requested Sorties

Result: Class Seniority First (Least Flexible Job)

Raw data was collected and compiled by week in a twenty-four week format. Figure 10 shows the raw data that was collected for the LFJ rule. The contents are the effective and minimum sorties for the four flights. The raw data format shows the level of variation in weekly sorties requests requirements. The variations in requests are due to each class's status and training requirements in the training schedule. Sorting the data by

training week number in the program makes the data easier to visualize. Figure 11 shows

the sorted effective sorties and the minimum required sorties.
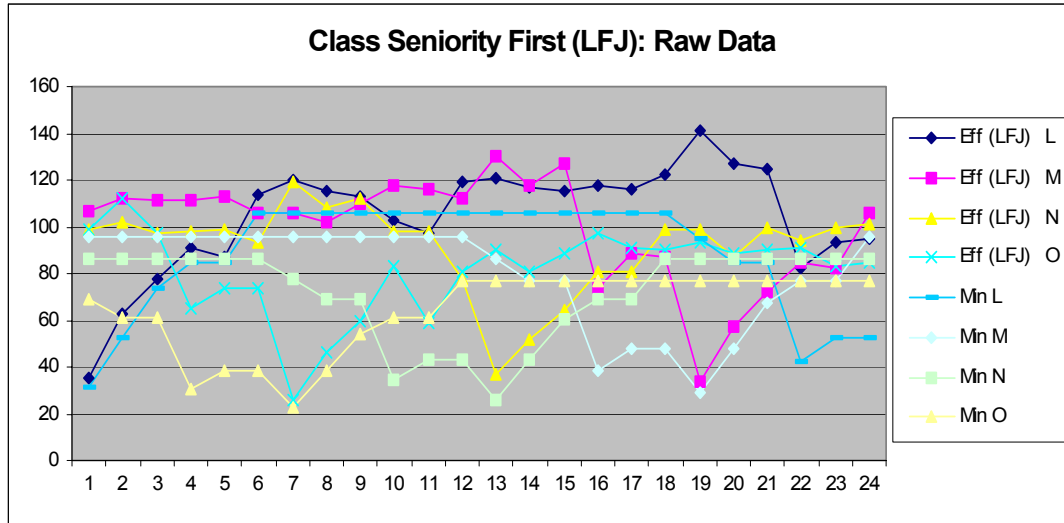
Figure 10. Class Seniority First (LFJ): Raw Sorties
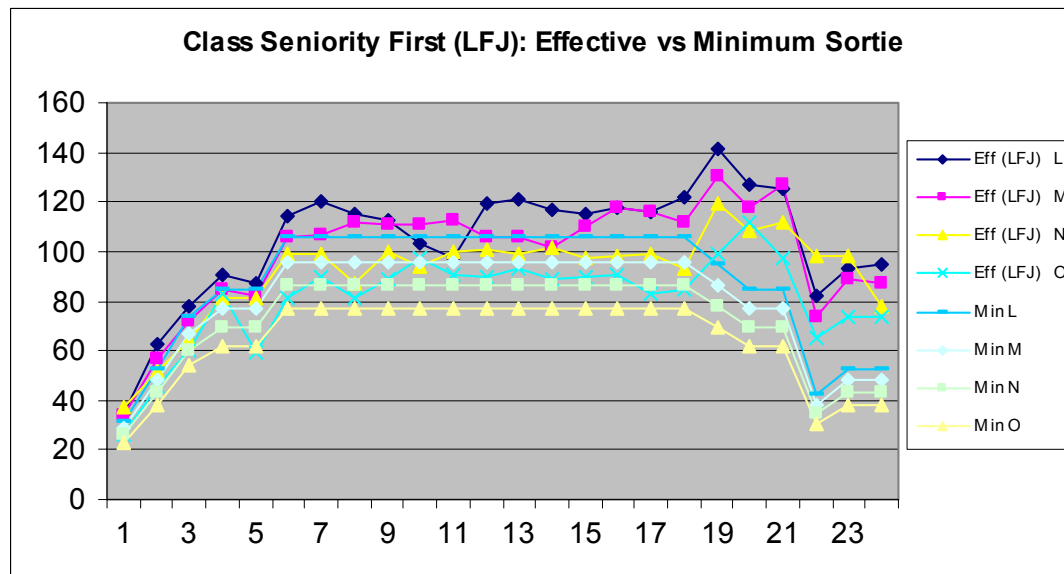
Figure 11. Class Seniority First (LFJ): Effective vs. Minimum

Information in Figure 11 can be separated further for clarity of information.

Figure 12 represents the minimum required sorties for each week. The minimum sorties

are calculated directly from the weekly requests, before the attrition model. The

minimum sorties data show the sorties requests for each flight are consistent with each

other. Figure 13 represents the sorties scheduled and the attrition model is applied to the

data. The sorties that failed through the attrition model are dropped from the statistic.

The sorties remaining are the effective sorties. The chart for the effective sorties shows

variation in caused by the attrition of the sorties.
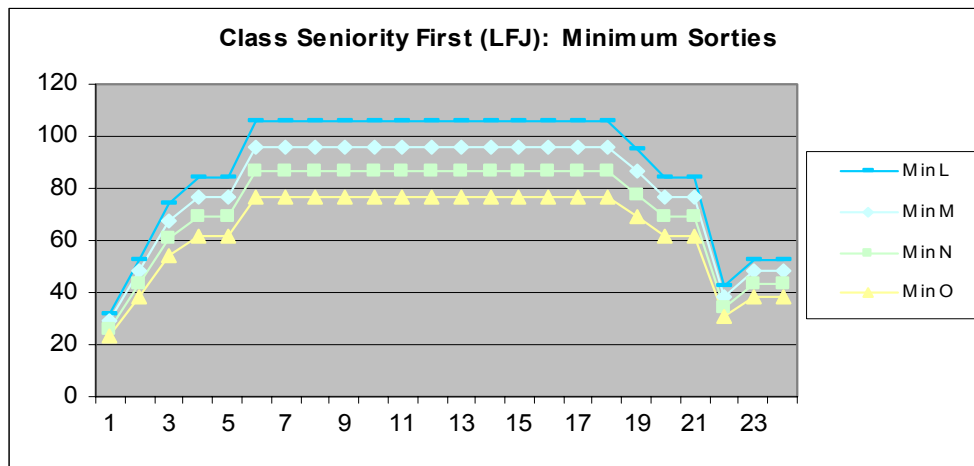
Figure 12. Class Seniority First (LFJ): Minimum Sorties
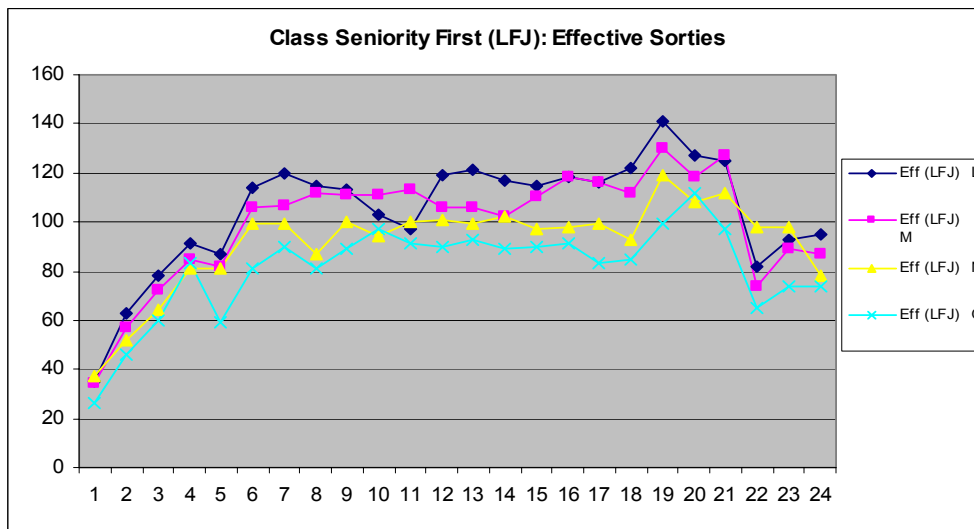
Figure 13. Class Seniority First (LFJ): Effective Sorties
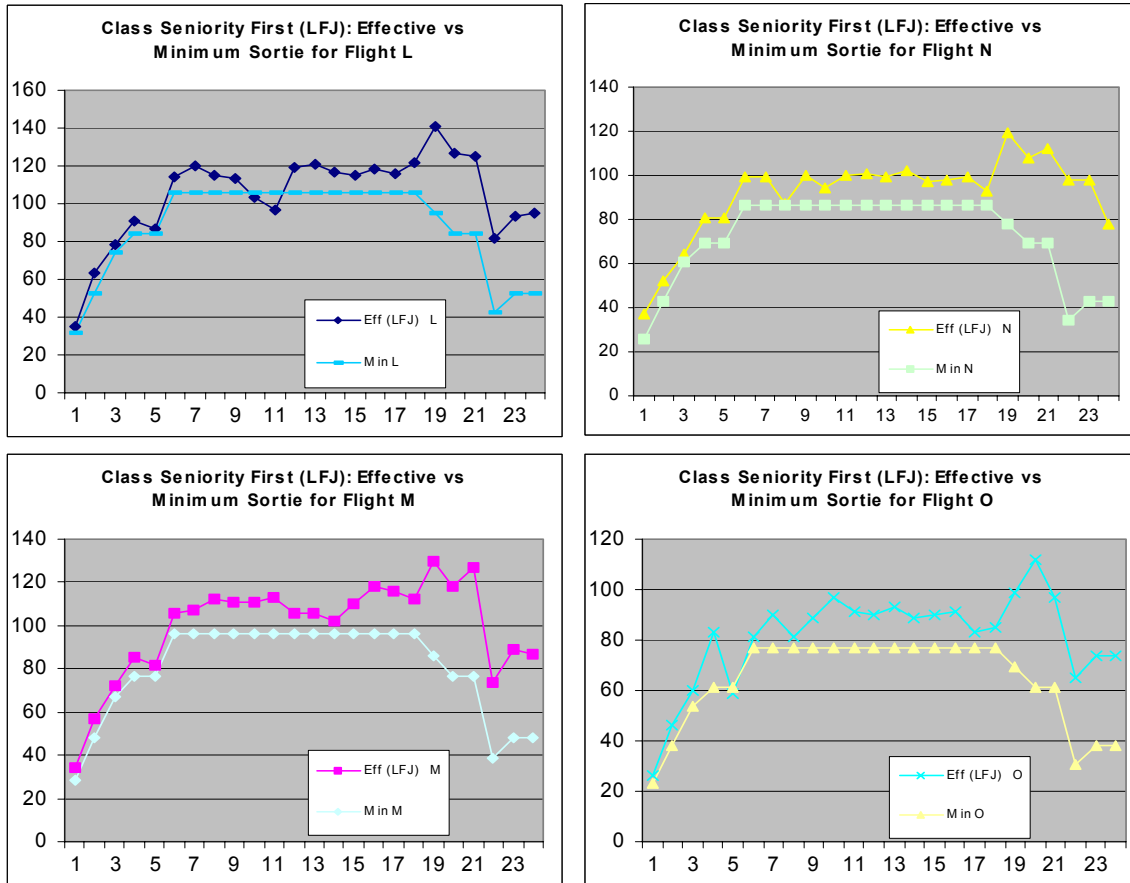
Figure 14. Class Seniority First (LFJ): Effective vs. Minimum by Flights

Next, statistics for individual flights are examined. Figure 14 shows the effective vs. minimum sorties broken down for each flight. Figure 15 shows a composite of the four charts in Figure 14. Weekly deviation statistics are also of interest to the squadron commander. Weekly deviation is measured by taking the weekly effective sorties and subtracting it from the weekly goal. Any positive or negative deviation shows either a gain or loss of hours or sorties in the training program. Figure 16 shows the weekly deviation sorties, broken down by flights, when scheduled under the LFJ scheduling rule.

The majority of deviations are positive, representing positive gain in training sorties for the training timeline.
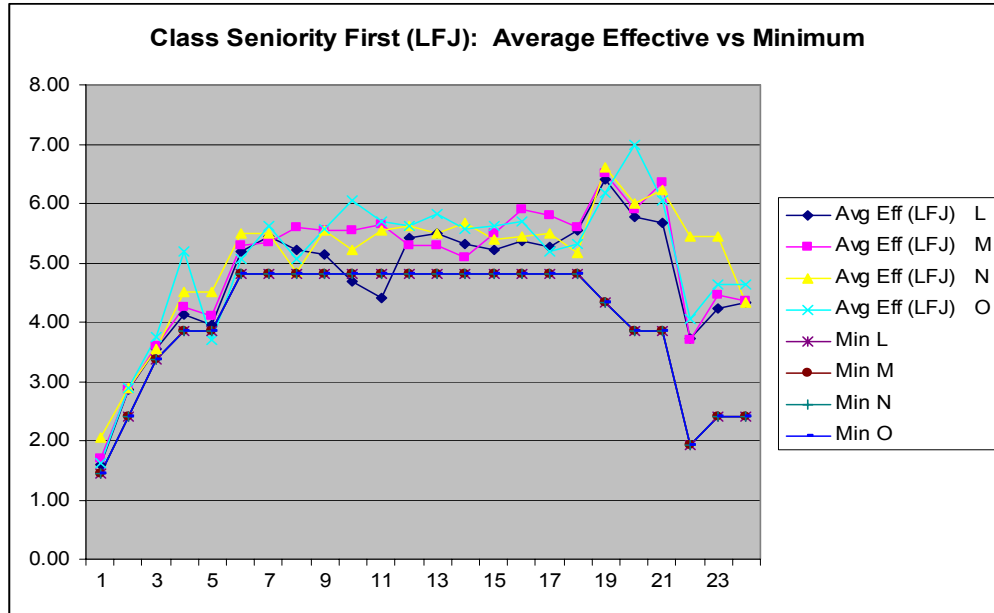
Figure 15. Class Seniority First (LFJ): Average Effective vs. Minimum
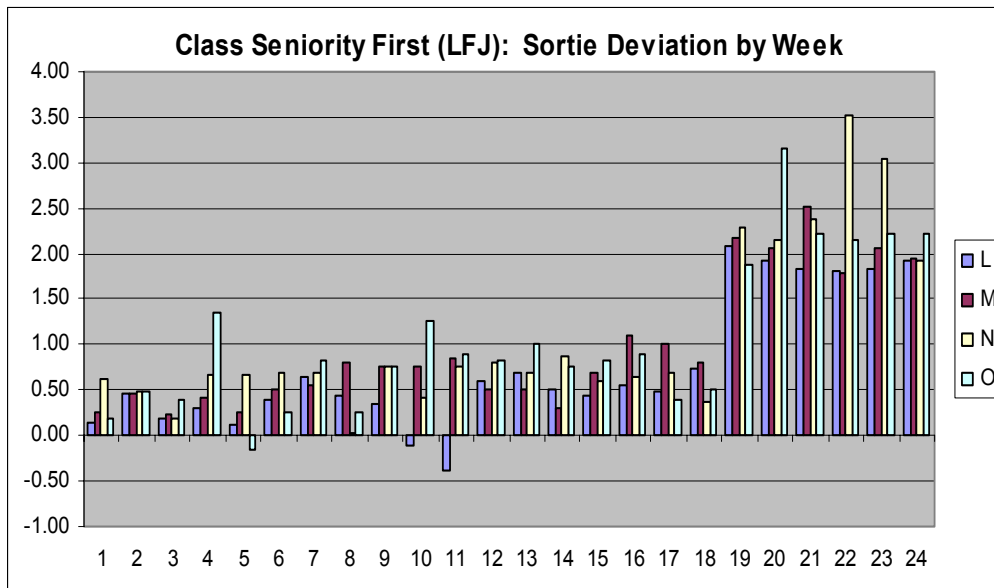
Figure 16. Class Seniority First (LFJ): Sortie Deviation by Week

Finally, will the scheduling dispatch rule allow the scheduler to generate a schedule that will allow the students to meet training requirements? Figure 17 shows both a weekly and a cumulative average effective sorties a student might expect to achieve through the training program. By taking the average effective sortie versus required sortie for each week, the scheduling dispatch rule can be measured for its effectiveness across the population of students. If the average effective sortie dips below the required sortie for any week, then the scheduling dispatch rule failed to generate a robust schedule. Figure 16 shows the Least Flexible Job dispatch rule, as applied to the Class Seniority First priority, is effective in generating a robust schedule. It shows an average student's expected number of completed training sorties to be above the required number as he proceeds through the training program. If the student continues to follow the schedule generated by the LFJ scheduling rule, at the end of the training period the student will successfully complete the training program and graduate. The charts show the scheduling dispatch rule generates a schedule that meets minimum sortie training requirements.
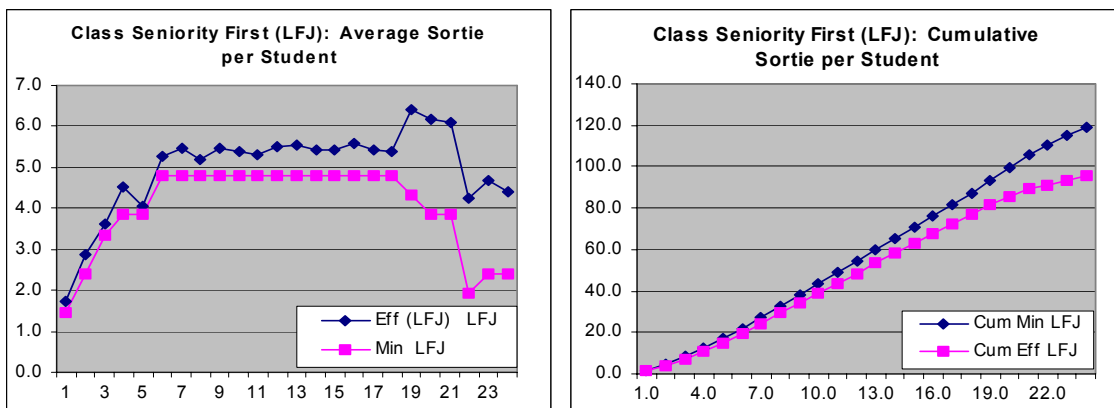
Figure 17. Class Seniority First (LFJ): Average/Cumulative Sortie per Student

Result: Largest Number of Requests (Longest Processing Time)

A flight with the largest number of requests can expect the longest time to complete training for the whole class for a given training event. It may sometimes be desirable to give this flight the highest priority to ensure all students in the class participate in training. If a student encounters a cancellation and needs to be re-scheduled, the flight that trains first in the morning receives the first re-scheduling opportunity available. This section contains the analysis of the schedule generated under the Largest Number of Requests priority, scheduling dispatch rule equivalent of Longest Processing Time (LPT) rule.
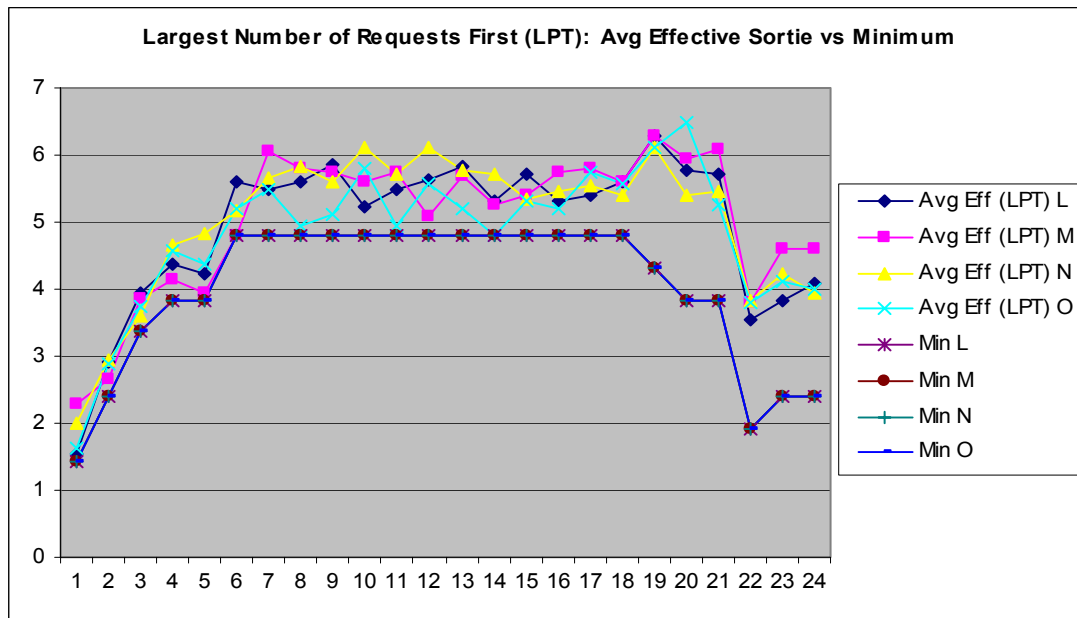
Figure 18. Largest Number of Requests (LPT): Average Effective vs. Minimum

The LPT rule is effective in generating schedules that have effective sortie rates that are consistently higher than the required rate. Figure 18 shows the average number of effective sorties per student for individual flight exceeds the average number of

required sorties. The higher effective sortie rate is consistent across the 24 weeks in the training program. Weekly deviation in Figure 19 shows all positive gain against required number of training sorties.
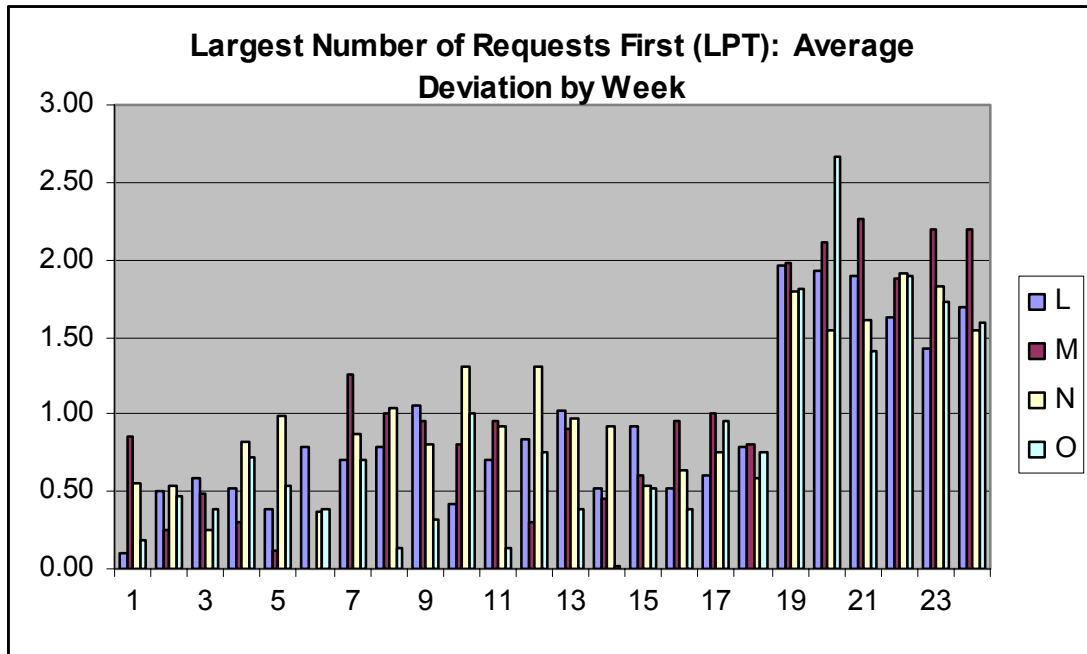
Figure 19. Largest Number of Requests (LPT): Average Weekly Deviation

Finally, will the scheduling dispatch rule allow the scheduler to generate a schedule that will allow the students to meet training requirements? Figure 20 shows the average number of effective sorties per student under this scheduling rule. The charts show that an average student can expect to meet and exceed weekly and cumulative training goals. The cumulative sorties of an average student under the Largest Number of Requests priority shows the dispatch rule is effective in generating a robust schedule. The cumulative effective sortie consistently remains above the required sortie. The

average student going through the training program under the schedule generated by the LPT dispatching rule can expect to successfully meet sortie requirements and graduate.
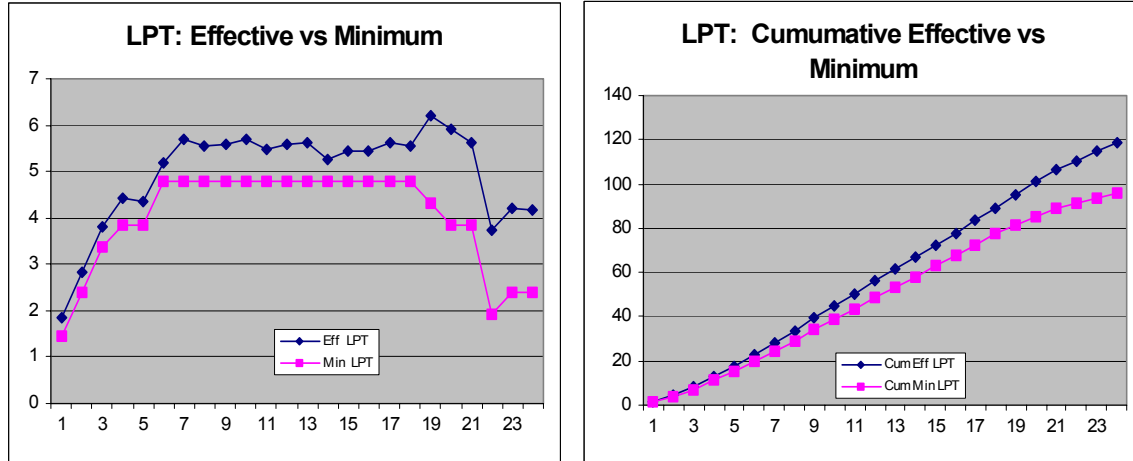
Figure 20. Largest Number of Requests (LPT): Average/Cumulative Effective vs. Minimum

Result: Flight Furthest Behind the Training Schedule (Minimum Slack)

The last scheduling dispatch rule used in this thesis is the Minimum Slack (MS) rule or its equivalent as the Flight Furthest Behind the Training Schedule priority. The scheduler might use this priority to bring a flight that is behind the required syllabus hours up to normal training status.

The average effective sortie rates for a student is above the required rate when the schedule is generated by the Minimum Slack dispatch rule. An average student can expect to have an effective sortie rate higher than the required weekly sortie rate for all four flights. Figure 21 shows the average effective rate is consistent throughout the 24-week training period.
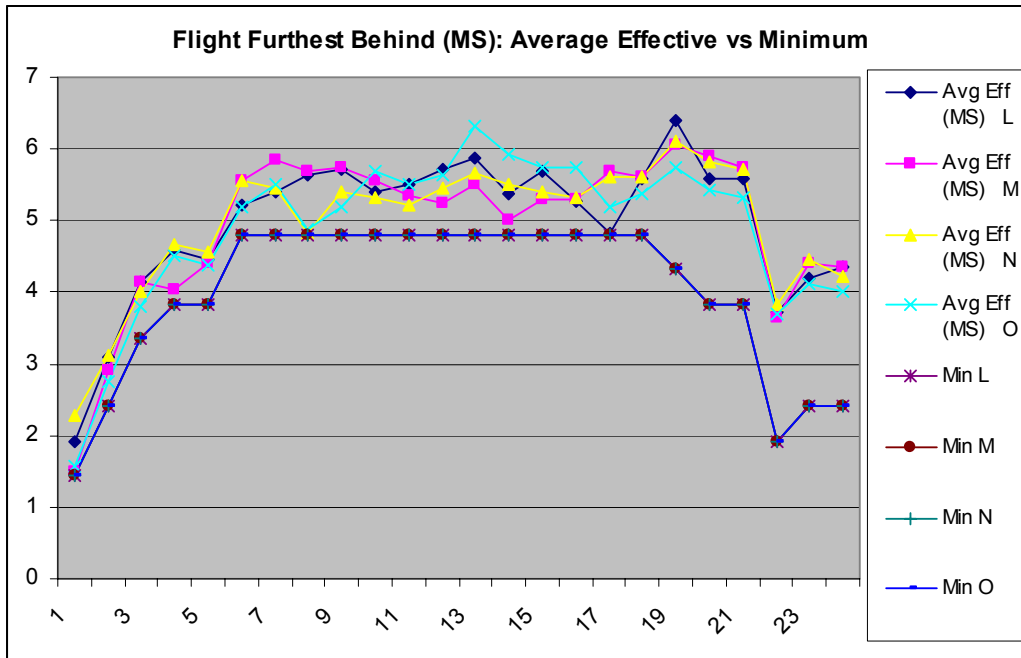
Figure 21. Flight Furthest Behind (MS): Average Effective vs. Minimum
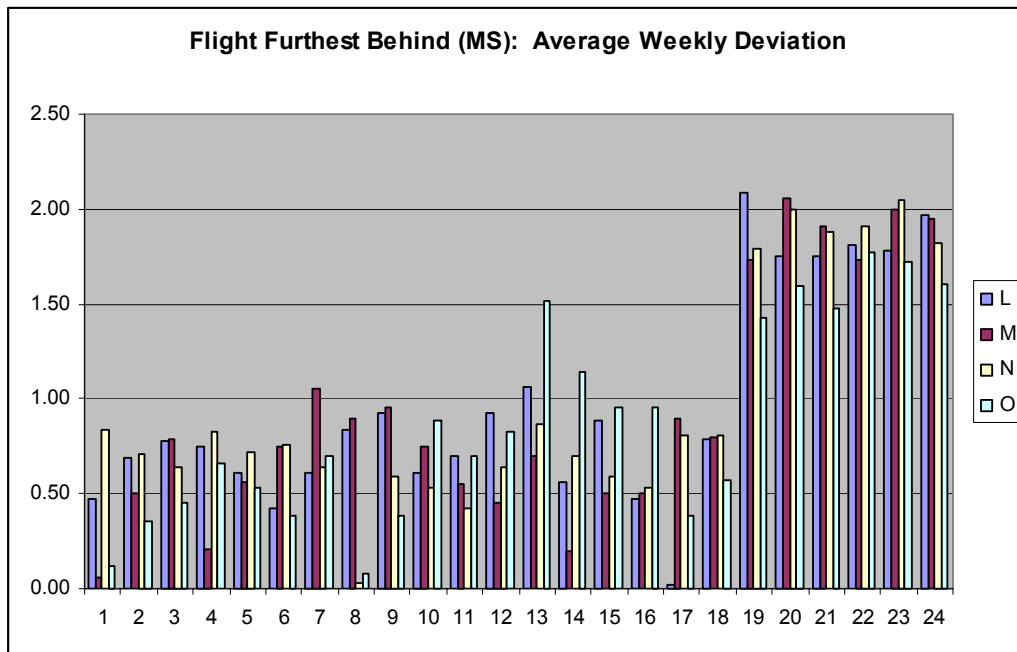
Figure 22. Flight Furthest Behind (MS): Weekly Sortie Deviation

The Minimum Slack dispatch rule also generates schedules with a positive weekly deviation. An average flight can expect to meet weekly sortie requirements under schedules generated by this dispatch rule.

The average and cumulative effective sortie shows the Minimum Slack dispatch rule is effective in generating robust schedules. Figure 23 shows both the weekly average effective sorties and the cumulative effective sorties are consistently higher than the required sorties. An average student training under schedules generated by the MS dispatch rule can expect to meet sortie training requirements and graduate from the training program.
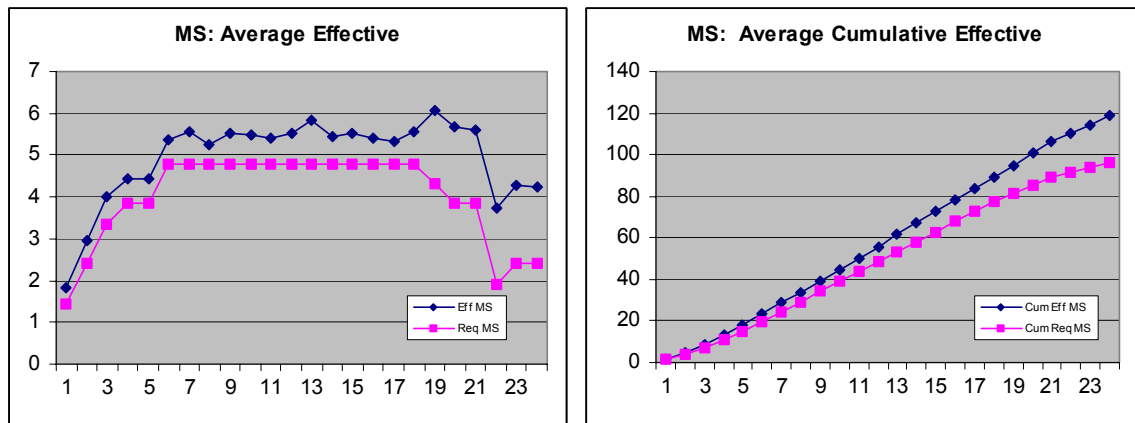
Figure 23. Flight Furthest Behind (MS): Cumulative Effective vs. Required

Comparison of Scheduling Rules

The analysis shows the three scheduling rules can generate robust schedules to meet the training requirements based on the notional training schedule. Figure 24 shows that all three scheduling rules generate schedules that produce effective sortie rates that are above the minimum weekly requirement. Furthermore, Figure 24 shows the

cumulative sorties are above the minimum sortie requirement of 96 sorties, as defined in

the training syllabus [AETC Syllabus, 2001, p. 1]. The interest shifts to how the

scheduling rules perform against one another. This section analyzes the performance of

the scheduling rules against one another.

To compare the rules, the average effective sortie rates of each of the scheduling

rules are subtracted, by corresponding training week, from the other two rates. The

difference in the value of the same training week indicates which scheduling rule

generates a schedule that has a better effective sortie rate for that week. The compilation

of the differences are graphed to highlight the differences in the values. The differences

in the sortie rates for the three scheduling rules was graphed. Figure 26 shows the

graphed differences for the effective sortie rates for the three rules.
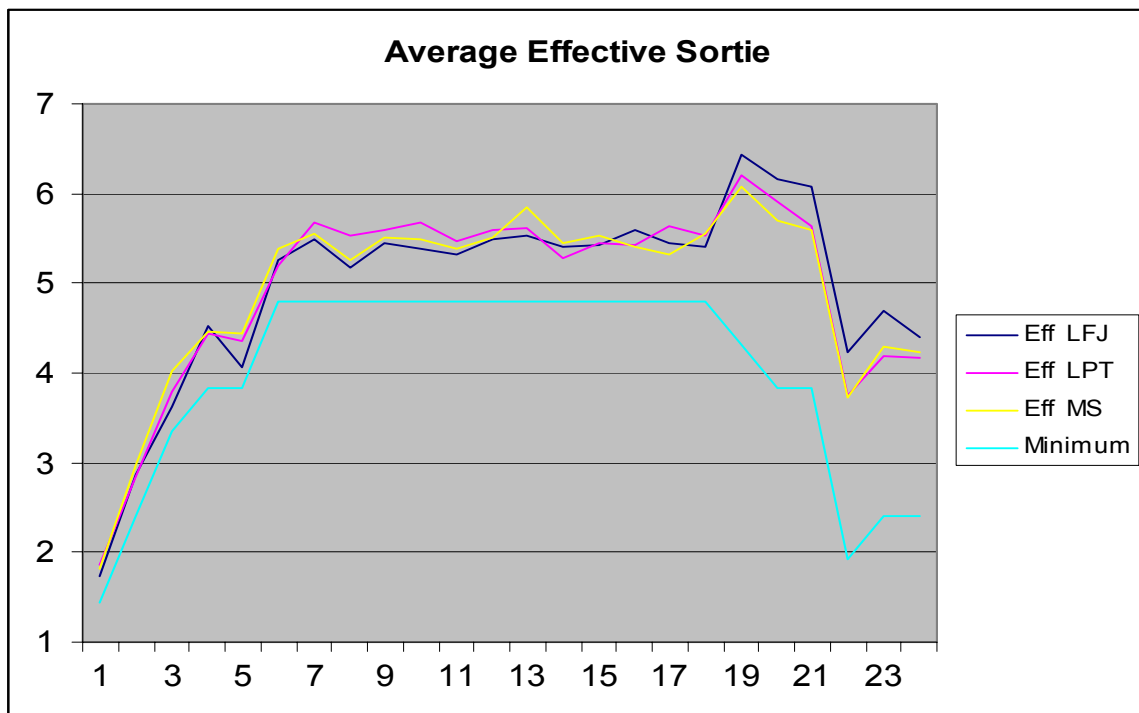
Figure 24. Scheduling Rules Average Effective Sortie
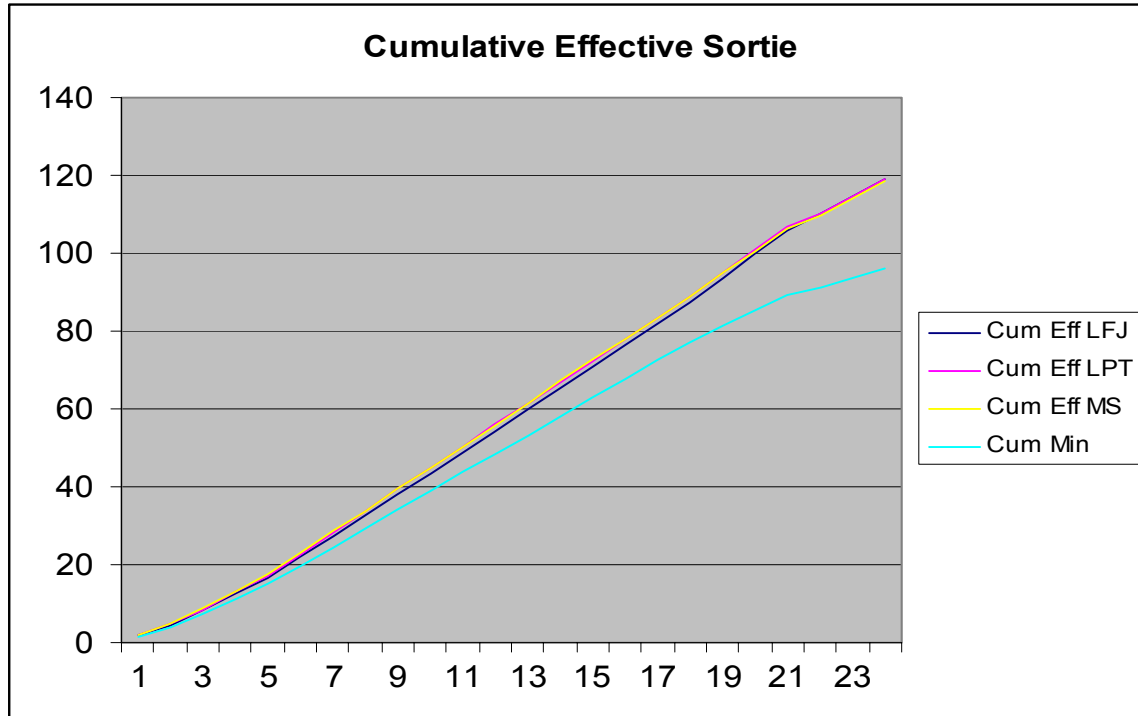
Figure 25. Scheduling Rules Cumulative Effective Sortie

Table 9. Cumulative Effective Sorties and Minimum

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Cum Eff LFJ	1.74	4.61	8.23	12.7	16.8	22.1	27.5	32.7	38.2	43.6	48.9	54.4	59.9	65.3	70.7	76.3	81.8	87.2	93.6	99.8	106	110	115	119
Cum Eff LPT	1.87	4.71	8.5	12.9	17.3	22.5	28.2	33.7	39.3	45	50.4	56	61.7	66.9	72.4	77.8	83.4	89	95.2	101	107	110	115	119
Cum Eff MS	1.81	4.78	8.8	13.3	17.7	23.1	28.6	33.9	39.4	44.9	50.3	55.8	61.6	67.1	72.6	78	83.4	88.9	95	101	106	110	114	119
Cum Min	1.44	3.84	7.2	11	14.9	19.7	24.5	29.3	34.1	38.9	43.7	48.5	53.3	58.1	62.9	67.7	72.5	77.3	81.6	85.4	89.3	91.2	93.6	96

The charts show the Least Flexible Job rule (LFJ) generates a schedule that has a lower average effective sortie rates at the beginning of the training program, when compared to the Longest Processing Time (LPT) or the Minimum Slack (MS) rules. This makes sense, because at the beginning of the training program the average flight has a lower priority than than other flights. As training progresses, the priority increases. The

effective sortie rate for the LFJ rule increases to eventually overtake the rate for the other

two rules. Figure 27 shows the cumulative rate for the LFJ as compared to the other two

scheduling rules. The effective sortie rate for the LFJ rule is behind for the majority of

the training program until the sharp increases at the 19th week begins to bring the

cumulative effective rate up. At the end, the cumulative scores are equivalent to the other

two rules.

Figure 26 shows both the LPT and the MS rules generate schedules that generally

have higher effective sortie rates than the LFJ rule during the first half of the training

program. The MS rule generates a higher effective sortie rate at the beginning of the

program, when compared to the LPT rule. The two rules are essentially equivalent in the

middle of the training program. The LPT rule eventually produces a higher effective

sortie rate than the MS rule at the end of the training program.
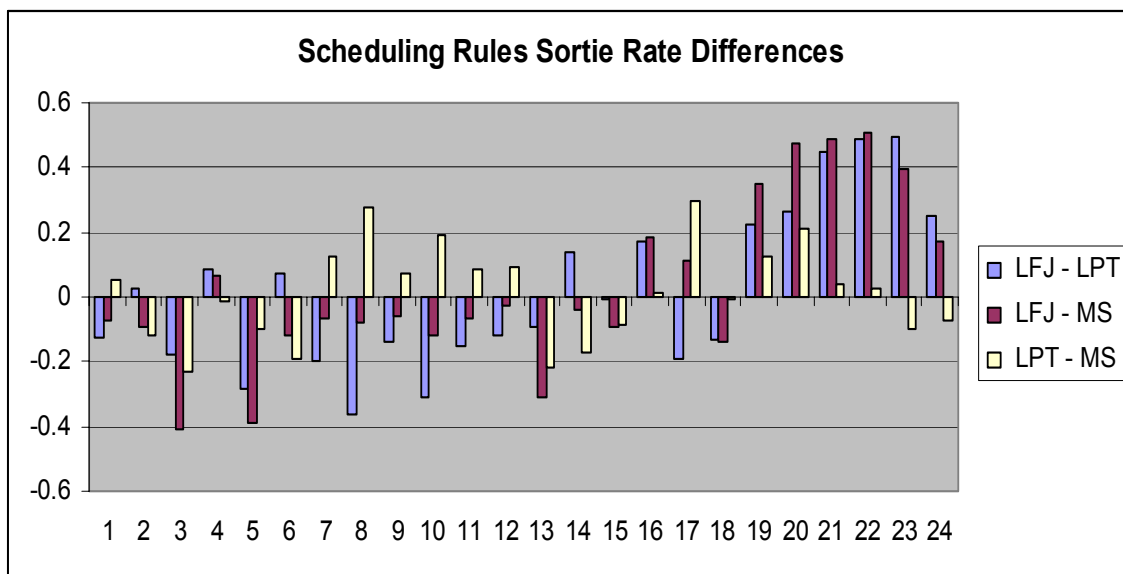
Figure 26. Scheduling Rules: Sortie Rate Differences
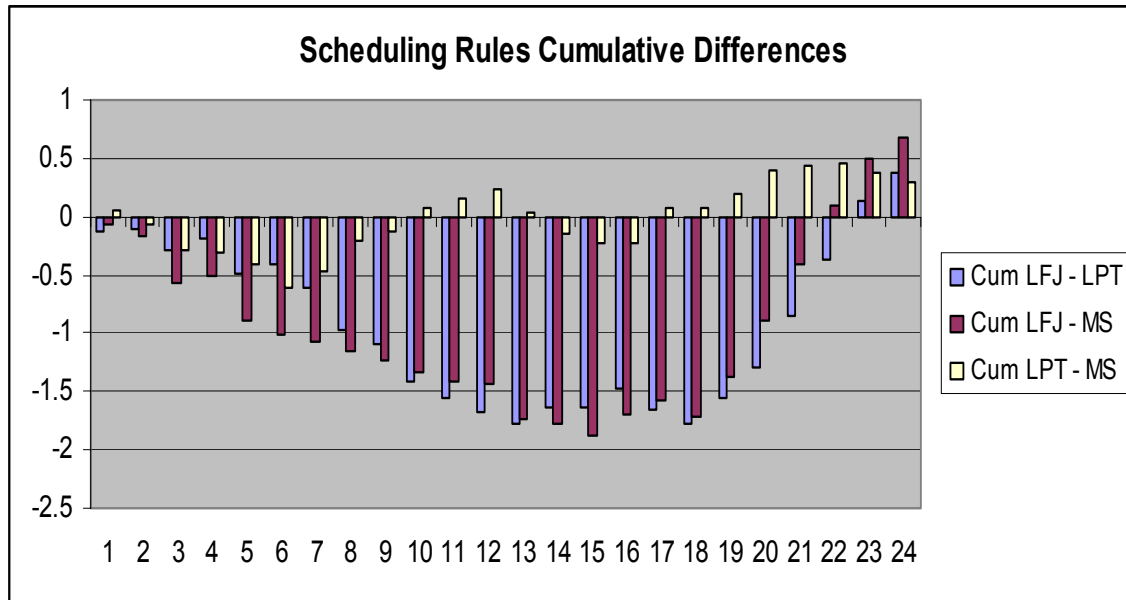
Figure 27. Scheduling Rules: Cumulative Differences

Summary of Scheduling Rules

The three dispatching rules, Least Flexible Job, Longest Processing Time, and Minimum Slack, corresponding to the flight priority of Class Seniority First, Largest Number of Requests, and Flight Furthest Behind, respectively, are effective in generating robust schedules. The three dispatch rules show a cumulative average sortie rate consistently higher than the minimum required graduation rate. Figure 25 shows the three dispatch rules produce similar results at the end of the training program; the differences are very minor. The dispatch rules might generate schedules with different flights receiving higher priorities, but the cumulative sortie rate remains essentially the same throughout the whole training program. The difference in the cumulative effective sortie rate is less than two sorties.

The differences in the effective sortie rates show there are instances where the rule is the most effective in generating a higher effective sortie rate. Suppose there exists a training program with all classes starting the training at the same time and progressing through the program at the same rate. Then the MS rule would be most beneficial to use at the beginning of the program, when compared to the LFJ and the LPT rules. The effective sortie rate is higher. The LPT rule is slightly better in the middle of the training program. This is due to the higher sortie requirement for each student, thus a larger flight will have more opportunities to fly, in turn generating higher effective sortie rates. The LFJ rule should be avoided throughout most of the training program because the effective sortie rate is lower until about the 19th week of the training program. From the 19th week on, the LFJ rule performs better than the other two rules. The higher sortie rates due to the extra cross-country sorties combined with a higher priority senior class generates a higher average effective sortie rate at the end of the training program. In this type of training environment, the three scheduling rules can be used. It should be noted, however, that the conclusions are limited to the small notional data set analyzed. A complete test of historical data should be considered before adopting any specific rule or rules.

In the 87th FTS training environment, the eight classes are all at different training levels. The scheduling rules take advantage of the environment and training level to produce a better effective sortie average. Because the classes are distributed evenly, the advantages are essentially nullified. Any differences in the effective sortie rates are due mainly to the randomness in the system.

In conclusion, each rule scheduled effectively in the simulated environment. In actual practice, it is anticipated that the squadron scheduler will use a mix of rules during a training cycle, selecting the rule most appropriate for the given operational setting. The SSDT allows the scheduler such freedom of action. By generating initial schedules for the scheduler according to current requirements, the SSDT frees the scheduler's time to attend to the details and variations that cannot be programmed.

Chapter V. Conclusions and Recommendations

This chapter reviews the importance of this research as well as the major issues covered during this research. The major findings of the pertinent literatures are summarized. This is followed by a review of the squadron scheduling environment and the description of the decision support system. The decision support system's scheduling algorithm is discussed. A review of the findings on the analysis of scheduling a notional training schedule is discussed. The chapter ends with a statement as to the importance of this research as well as recommendations for future research relating to this topic.

Background

The scheduling process at the 87th FTS exhibits similar characteristics to known manufacturing systems. Job orders, classes, simulator sorties, and flying sorties have processing times and due dates. Resources such as the T-38 training aircraft and Instructor Pilots are non-depleted resources. Detailed scheduling of the students and class are required to ensure all required students are available for certain class lectures to avoid duplication of efforts. Bad weather and aircraft breakdowns may unexpectedly occur reducing the available sortie times. The requirement to meet the minimum sortie training hours may be impacted by these unexpected events.

Squadron scheduling directly influences every pilot's life in the squadron. The scheduling determines when the pilots attend classes, fly sorties in the simulator pods, or when they fly the aircraft to gain valuable skills and experience. Squadron scheduling generates schedules to place students to these activities.

To graduate from fighter pilot training, students must meet minimum hours in both simulator training and flying sorties. In addition to the hours, students must also pass all exams from the class lectures. To meet these requirements, the squadron scheduling shop develops draft schedules for class lectures, simulator sorties, and flying sorties. Class lectures are required to introduce students to appropriate concepts. Once the students learn the lecture material and pass examination, the student attends simulator training for the appropriate material. After simulator training is passed, the student may begin flying sorties for the appropriate material already covered.

The main goals for the squadron's scheduling effort is to meet certain sorties or flying hour goals, balancing the requirements of the four different flights, and maintaining balance in the students' class work and simulator training. The squadron scheduler prepares a schedule to meet the flying goals based on the availability of air traffic controller, student, IPs, weather, night sorties, prerequisites, and other factors.

Squadron scheduling must consider minimum hourly goals/requirements of the students. Meeting the student's hourly goals requires three different schedules, the academic schedule, scheduling classrooms for in-class lectures, the simulator schedule, provide students with cockpit environment training without leaving the ground, and aircraft sortie schedule, allocate students and IPs to available aircraft .

One area of interest is how to create schedules in the face of different and sometimes conflicting objectives. Students want to maximize their training hours, flying as much as possible. Maintenance desires as few aircraft utilized to minimize on the cost of time and manpower. Sometimes sorties must be cancelled or delayed due to outside or

uncontrollable factors such as visiting dignitaries, weather, or other factors. One question is how to create a robust schedule that will still satisfy the training objectives.

Belton and Elder's Visual Interactive Modeling (VIM) uses expert knowledge to guide the schedule generation process. It utilize an interface to some heuristic engine, with a built-in control mechanism, to influence heuristic search, preference, or performance criteria [Belton and Elder., 1996, p.162]. This approach was effectively adopted in developing the SSDST for the 87th FTS. It provides structure and support, while allowing the squadron scheduler the ability to interact with the schedule, providing intangible expertise that cannot be programmed.

Visual Interactive Modeling and Scheduling

Within the VIM framework, the squadron scheduler interacts with the software throughout the scheduling process. The squadron scheduler generates an initial schedule by picking the desired scheduling rules based on certain priorities. If changes are needed, the squadron scheduler generates subsequent schedules by modifying the initial schedule. The process repeats until a robust schedule is produced.

Scheduling Dispatch Rules

The schedules generated in the squadron scheduling shop are deterministic schedules generated and finalized one week in advance due to contracts required with the maintenance and simulator shops. Using general dispatch rules, the squadron scheduler can quickly create initial feasible schedules, and through the VIM process iteratively modify the initial schedule to arrive at schedules that satisfy the scheduler's requirements.

Software Design and Implementation

Software design and implementation takes advantage of existing tools/software, to speed up the scheduling process. The existing MS Excel scheduling tool are familiar to schedulers and commanders and has been used to create past schedules. The existing spreadsheet already possesses the required official format to the different outputs for the schedules and contracts.

Notional Training Schedule

A notional training schedule is developed to test the scheduling dispatch rules and its implementation in the software. The notional training schedule is based on the 120-day training program duration.

Attrition Model

An attrition model is used to simulate the different attrition types and percentages. The attrition model is used to simulate the attrition of sorties that can typically be found in a training environment. Once the schedule is generated, a post-scheduling attrition model is applied to the schedule during statistical analysis.

Physical and software performance

The software interface provides a user-friendly environment. The scheduler chooses the desired function by pushing appropriate buttons to activate the codes. Data is entered directly into the spreadsheet. Once the data is entered and the appropriate priority button is pushed, the scheduling algorithms build a draft schedule.

Statistical Analysis and Performance of the Scheduling Dispatch Rules

The three dispatch rules being measured for the analysis are the Least Flexible Job, Longest Processing Time, and Minimum Slack corresponding to the flight scheduling priorities of Class Seniority First, Largest Number of Requests First, and Flight Furthest Behind, respectively.

Result: Class Seniority First (Least Flexible Job)

The Least Flexible Job, as applied to the Class Seniority First priority, is effective in generating a robust schedule. The weekly deviations for the Class Seniority First priority, broken down by flights, are mainly positive, representing positive hours above the training requirements. The average student's cumulative training hours under the schedules generated by the Least Flexible Job dispatching rule shows the dispatch rule is effective. An average student can expect to complete more hours than the minimum required as the student proceeds through the training program.

Result: Largest Number of Requests (Longest Processing Time)

The Largest Number of Requests priority is effective in generating schedules that have an effective sortie rate consistently higher than the required rate. The average number of effective sorties per student across the student population is higher than the average number of required sorties. It effectively generates a robust schedule that keeps the student sortie rate of the majority of the students above the requirement. The average student going through the training program under the schedule generated by the LPT dispatching rule can expect to successfully meet sortie requirements and graduate.

74

Result: Flight Furthest Behind the Training Schedule (Minimum Slack)

The Minimum Slack dispatch rule is effective in generating effective sortie schedules. Average Effective sortie rates for a student across the student population is above the required rate when the schedule is generated by the Flight Furthest Behind priority or Minimum Slack dispatch rule. The Minimum Slack dispatch rule also generates schedules with a weekly deviation that is positive. The four flights show a consistent positive average deviation trend. The cumulative effective sortie is consistently higher than the required sortie. An average student training under schedules generated by the MS dispatch rule can expect to meet sortie training requirements and graduate from the training program.

Summary of Scheduling Rules Finding

The three dispatching rules, Least Flexible Job, Longest Processing Time, and Minimum Slack, corresponding to the flight priority of Class Seniority First, Largest Number of Requests, and Flight Furthest Behind the Training Schedule, respectively, are effective in generating robust schedules. The three dispatch rules show a cumulative average sortie rate consistently higher than the required graduation rate.

Recommendations for Future Research

A key result of this research effort was the successful implementation of the VIM concept to create robust schedules using scheduling dispatch rules. Through the VIM concept, the scheduler interacts with the interface to make iterative adjustments to the initial schedule to arrive at a desired schedule. There are several areas that need more research.

One of these areas is the attrition model. The author was unsuccessful in tracking down the source of the attrition rates, how it was specified, and whether the rates are valid or not. A better attrition model can contribute to a more accurate prediction in sortie attritions. The attrition model can also improve by finding a better distribution model than the uniform distribution used in this thesis.

Another research area relates to the requirements. A good requirement model will provide a smooth training program for both the students and the scheduler. A good requirement model will give the scheduler a good forecast of the required sorties for future scheduling.

Further work can also be done on the visual interface. The modified Excel tool has some great features that show when a schedule is busy and when it is free. A better visual interface can show the busy status for each flight, when a time slot is free for re-scheduling opportunities, and also show if any constraints are liable to be broken.

Further research can be done on other scheduling dispatch rules or combinations of rules. Other rules might yield better results than the three used in this thesis. These rules should be run multiple times to collect better data than the single run for each rule, as used in this thesis.

Last, this thesis was conducted on a non-standard scheduling tool. AETC is adopting TIMS and are currently installing it at the different training bases. All training squadrons are required to use TIMS once it is operational. The TIMS scheduling methodology is still essentially a manual scheduling method similar to the current manual method. This thesis has shown that scheduling rules can be employed that will speed up the schedule generation process and provide robust schedules. TIMS has some

experimental scheduling rules that were researched but are currently not implemented.

Future research on implementing scheduling rules in TIMS is highly recommended.

Appendix A. Squadron Scheduling Terms

This appendix lists the different scheduling terms in the fighter pilot training and scheduling environment. This is a condensed list of the terms. More detailed explanations of the terms can be found in the AETC training syllabus.

Air Traffic Controller (ATC).

The squadron utilizes both in-house air traffic controller (ATC) (performed by the squadron's IPs), and Laughlin's flight line ATC. The decision of which type of ATC to assign to particular sorties is made by the Chief Squadron Scheduler. This decision depends on such factors as the availability of the squadron ATC, flight requirements, and the work tempo. Squadron ATCs are experienced IPs and ATCs. Flight line ATCs are generally student ATCs being trained in tandem with student pilots at Laughlin AFB. In some situations, flight line ATCs are less effective at directing traffic. Squadron ATCs are also IPs, thus when sorties requirements utilized the squadron ATC, the flight line ATCs are used. When tempo or maneuvers require experienced ATCs, the squadron ATCs are used.

Student.

Students go through a host of training at the 87th Flying Training Wing. Students must be meet specific requirements and milestones to be available to fly sorties. Students must pass coursework, simulator rides, ground checks, and specific prerequisites for certain types of rides. IPs must certify the student is ready. If a student fails at one or more requirements or simulators, the student might be pulled from flying availability until he/she passes the requirements. IPs can pass or fail a student for certain sorties

based on flight and simulator performances, and the IPs evaluation of the student's readiness.

Instructor Pilots (IPs).

Instructor pilots are required for certain qualified check rides and all tandem rides. In addition, IPs are sometimes required to fly solo in a formation to evaluate solo students' formation abilities. IPs also require solo sorties to keep their own abilities fresh.

Student-IP Match.

A student is usually assigned to a primary IP and a secondary IP. This gives the student a better understanding of the IP and his instructions. If the primary IP is not available, then the secondary IP can instruct the student. If both the primary and secondary IP is not available, the flight commander can assign an available IP to instruct the student.

Weather.

Southwest Texas has stable good weather most of the year. However, at certain times of the year have inclement weather that may affect flying availability. Fog and low cloud may prevent certain check rides due to poor visibility; IPs cannot verify student's aerial maneuvers, and prevention certain maneuvers, and so forth. Heat is a critical factor during the summer, often in the afternoon, when the temperature can be over 110 degrees Fahrenheit. Heat can induce loss of power in aircraft and heat exhaustion in the pilots. At certain time of the summer, the heat can ground all aircraft during the afternoon for weeks.

Night Sorties.

Night sorties are usually scheduled once per month, on the first Tuesday of each month. The twelve hours rest rule still applies. A student or IP must still have twelve hours of rest before training can resume. In addition to the night sortie, students can only attend one other event (day sortie, course work, or SIM training) in addition to the night sortie.

Prerequisites.

Pre-requites are requirements that must be satisfactorily completed before the present sortie can be scheduled. Failing a prerequisite does not necessarily prevent a student from flying a sortie. However, the flight commander must approve student for availability, and the student must pass the required rides before progressing further in the program.

Other Factors.

Other factors must also be taken into consideration when making the schedule. Other schedules, such as simulator training schedule and class schedule, can affect the sortie scheduling. Classes are given in blocks during the day. Simulator training can be done whenever the students (and if required IPs) are availability. Holidays, leave, sick leave, and other general factors can also affect the availability of the students, IPs, ATCs, and other resources.

Flight Level Schedule.

Flight schedulers must prepare a schedule for each student in the flight based on availability of IPs, student, class work, simulator training, and the student's training status.

Instructor Pilot Availability.

Instructor pilots are required for most of the first half of the training program, when students have not gained enough proficiency for solo flying. As the training program progresses, students fly more solo missions. Primary IPs sometimes are not available, so the secondary IP will take their place to train the student. If the primary and secondary IP are both not available, the flight commander can assign another IP to train the student.

Student Availability.

Students sometimes are not available due to failing prerequisites, being ill, or are disqualified for not meeting the rest requirements. The prerequisites include passing exams, simulator training, other check rides, and instrumentation. Student must also demonstrate ability to the IP that the student can perform the training sortie before being allowed to fly. Even if the student meets all requirements, if the IP feels the student not ready, the student receives further one-on-one training before he or she is allowed to fly.

Class Work.

Class work includes written and oral work and exams. The work must be completed and the exam passed before the student can be available to fly. The flight or squadron commander can approve the student for flying, even if the student does not pass. However, this is rarely done. Students failing class work must retake the failed exam until the failed students pass or otherwise satisfy some requirement.

Simulator Training.

Simulator training familiarizes students to the aircraft, instruments, procedures, and maneuvers performed, before the student can fly on the aircraft. Students are

required to pass certain simulator training before he can proceed on to aircraft sorties. Simulator trainings are scheduled for certain specific times. Students can also gain opportunity training when the simulator is not scheduled or in use.

Student's Status.

The student's status in the training program also play a deciding factor in the flight scheduling process. If a student is ahead of the program in terms of average hours flown, course work, or other requirements, the flight scheduler may place the student at the bottom of the priority list for flying sortie. If a student is behind in flying hours, needing to pass check rides, or other milestones that cause him to fall behind the training schedule, he receives higher priority when determining student sorties. All students are encouraged to accumulate additional flying time when feasible.

Re-scheduling

Both the squadron and flight scheduler perform re-scheduling when there is a change in the schedule. The squadron scheduler performs the re-schedule when events affect the squadron as a whole. The flight scheduler performs the re-schedule when events only affect the flight.

Squadron Re-scheduling.

The squadron scheduler re-schedules when certain events affect the whole squadron. Flights fly sorties in waves throughout the day. If an event affects part or a whole day, grounding aircraft, or precluding aircraft from taking off or landing, parts or all flights might miss the whole day's worth of training hours. They will then need to make up the hours later during the week. Special events such as visiting VIPs and unannounced requirements also require re-scheduling. Unforeseen weather such as

sudden thunderstorm, wind, heat, and other weather conditions, can prevent aircraft from taking off for part of a day or longer if the conditions are persistent.

Flight Re-scheduling.

Flight schedulers will also re-schedules at certain times. Some students may fail certain events (qualifying sortie, simulator training, exams) and are not qualified for the next sortie in the training program. Schedulers may schedule another student in this student's place. Students can fall behind in flying hours or need to pass previously failed events. Other students may pull ahead of the training program, thus not needing as many training hours in a given week. Some events needed to be cancelled, delayed, or swapped with other events to meet the educational goals. These events also needed to be re-scheduled. All the events that do not affect other flights or the squadron as a whole, but still require re-scheduling are done within the flight.

Work Rules

Several work rules are presented. The 12 hours work rules are mandatory for everyone. The 2-combination work rule applies only to student pilots. Discussions of these work rules are discussed below.

Twelve hours cycle.

The work rules are set by AETC on a maximum 12 hours cycle. Student pilots and Instructor Pilots (IPs) log by flight logs, simulator logs, class time logs, or visually at the squadron and flights. By checking the logs, squadron commanders can verify if the pilots' workloads stayed under the requirements. The pilot cannot resume training on flying training, simulator training, or classroom work until he/she has rested for at least twelve hours.

For example: a student logged out of a simulator training at 18:02 the previous day. He has a 0600 check rides scheduled for the next day. Since he had only 11 hours 58 minutes of crew rest, his commander denied his check ride until he has met the required 12 hours rest.

Two Combination Work Rule.

Students cannot fly more than two flights back to back in a 12 hour day. The combination includes Sortie-Sortie, Sorties-Simulator, Simulator-Sortie, and Simulator-Simulator.

Sorties Definition

Student and IPs fly several different types of sorties:

Local Sorties.

These sorties are flown in the local area. These sorties are flown for check rides such as contact, instrumentation, formation, navigation, solo, and others.

Off-Station Sorties.

Flown during the weekend. The student and IPs take the aircrafts off-base for one or two days for cross-country training. The aircraft would use the destination's facilities, including maintenance, fuel, and other supports and facilities. Once the training is complete, the aircraft return to Laughlin to be prepared for the next day.

Night Sorties.

Night sorties are generated for training in nighttime flying. Night sorties occur usually the first Tuesday of the month, once per month.

Statistics

The squadron gathers statistics on each of the sorties flown each day. After completing or aborting a sortie, each student and/or IP must immediately record the sortie's information into the flight database before he can proceed with the day's activities. The flight database thus accumulates all the information available. At the beginning of each week, statistics from the previous week are compiled and reported to the squadron commander. The month's statistics are reported to the wing commander.

The statistics currently being reported to the squadron and wing commanders are the same. They include: Student's flying status, status of students on the watch list, flying hours goal for the month, deviations from the plan for the month, and a daily plan for the week. The briefing slides and statistics are standardized throughout the wing.

Student's Flying Status.

The T-38 SUPT Student Status shows the current individual flights' status. It summarizes the size of the student body, any cap, any student placed on administrative hold, the current status of the flight's average flying/SIM hours versus the month's goal hours, and any relevant remarks.

Students on the Watch List.

The T-38 Commander's Awareness Program (CAP) Summary report summarizes the flights' students that are having trouble with the training. It lists the flight, type, progress, trig, the next check to pass, and any relevant remarks. This information is used by the commander and the squadron scheduler to watch students who are behind the training schedule and give preference to training sorties when possible.

Flying Hours Goal for the Month.

The monthly goal slide provides statistics for the month beginning from the first day of the month. The information contains the accumulated flying hours attained for the month as well as the goal flying hours. It shows any deviation of hours from the month's goal, in number of days (example: 1.48 days behind). It also shows the contracted flying hours for the year, and any deviation from the contract.

Deviations for the Month.

The deviations information summarizes any additions or cancellations to aircraft for that month. It compares that statistics versus the historical average. The categories contained in the slides are operational deviations, weather deviations, maintenance deviations, and other deviations. These are sorties being added or cancelled due to the respective reasons.

Daily Plan for the Week.

The Daily Plan shows the squadron's daily events for the week. It gives the details on the number of day and night sorties planned, cross country sorties, number of aircraft that are off station due to cross country sorties or maintenance problems, total scheduled sorties for the each day, any expected attrition (students not meeting the sortie requirement, thus failing the sortie and must retake again), expected effective number of sorties flown, number of sorties required to meet monthly goal, any gain or loss in number of days, number of scheduled aircraft to be used, time window for first take-off and last landing.

Squadron Database.

The squadron database stores all flight data. Students and IPs must enter the data for their sortie immediately after their completion or abort, before they can continue with their daily task. This ensures the data is accurately captured for each sortie, and will provide an accurate account of the weekly progress. As a caveat, the data is only as accurate as the person entering the data. The data is stored and reported weekly to the maintenance shop, the squadron commander, and the wing commander. Pertinent information is used for the planning of future schedules.

Squadron Scheduling Excel Sheets.

The squadron scheduling shop uses several spreadsheets as tools to keep track of their statistics and generate new schedules. The Spreadsheets are Excel sheets that are formatted to generate the schedule. Data are entered into relevant fields where formulas in other cells automatically calculate and output data into relevant cells in the schedule.

Jet Order.

The Jet Order lists the number of sorties needed for the week. The Jets Order portion is submitted to the maintenance squadron with a minimum one-week prior. The squadron confirms the order the Thursday of the prior week for the number of aircraft. The SIM Order portion is submitted to the SIM squadron with a minimum one-week prior. The squadron confirms the order the Thursday prior.

Build Jet Schedule.

The Build Jet Schedule sheet prints out the schedule for each day of the week. The schedules are built by hand, one line at a time. The T/O Interval describes the interval of time between each sortie. The Cap is the number of aircraft needed for that

day. Jet turn is the turn time for each jet. Sorties are the number of sorties scheduled to fly for that day. This setup is repeated for each day of the week.

Build Simulator Schedule.

The Build Simulator Schedule generates the schedules for the simulator training. The schedule lists individual time slots for each flight. The data is manually entered by hand into each slot. The example shows in Monday. The capital letters L, M, N, and O are the individual flights in the 87th FTS. This schedule is repeated for the other days of the week.

Academic Scheduling.

The Academic Scheduling form generates the academic schedule for the week. The fields include class titles, instructors, room number, star and finished time, which flight assigned to which location, and the type of classes.

Appendix B. Notional Daily Training Schedule

This appendix lists the notional training schedule. The list consists of the daily training requirements for each flight. The daily requirements are further broken down into individual Go requests. Tuesday's night flying training are also included. Weekend cross-country flights are included in the Wednesday/Thursday/Friday in the Go 4 column. This is done because the scheduling format does not include Saturday and Sunday. So the cross-country requests for Saturday and Sunday are moved into Wednesday and Thursay's schedule. This arrangement does not affect the original schedule in any way.

Table 10. Notional Detailed Daily Sortie Training Schedule

	Daily Sortie Schedule																			
1	**Monday**				**Tuesday**				**Wednesday**				**Thursday**				**Friday**			
L / LO	6	3			6	3			6	4			6				6			
M / MO	8	8	8		10	10	8		10	10	8		8	8	8		8	8		
N / NO	8	8	8		8	8	8		10	10	8		8	6	6		6	6		
O / OO	6	6	6		6	6	6		8	8	6	**8**	6	6	6	**8**	6	5		**8**
2			**75**				**79**				**113**				**93**				**70**	**379**
L / LO	8	8			8	8			9	9			9				8			
M / MO	8	8	8		10	10	8		10	10	8		8	8	8		8	8		
N / NO	8	8	8		8	8	8		10	10	8		8	6	6		6	6		
O / OO	6	6	4			4	4	**8**	8	8	4	**8**	8	7		**8**	5	5		**8**
3			**80**				**84**				**119**				**93**				**71**	**396**

89

Block 1 (row id 4)

Row																				
L	7	7	7		7	7	7		8	8	8		6	6	5		5	5		
M	8	8	8		10	10	8		10	10	8		8	8	8		6	6	4	
N	8	8	8		8	8	8		10	10	8		8	6	6		6	6		
O	6	6	4		6	6	4		8	6	6	**8**	8	7		**8**	5	5		**8**

Group totals: 85 | 89 | 125 | 101 | 73 — Total **422**

Block 2 (row id 5)

Row																				
L	8	8	7		8	8	7		10	8	8		8	6	6		7	7		
M	8	8	8		10	10	8		10	10	8		8	8	8		6	6	4	
N	8	8	8		8	8	8		10	10	8		8	6	6		6	6		
O	8				8				5	5		**8**	8			**8**	5			**8**

Group totals: 79 | 83 | 118 | 98 | 73 — Total **397**

Block 3 (row id 6)

Row																				
L	8	8	7		8	8	7		10	8	8		8	6	6		7	7		
M	8	8	8		10	10	8		10	10	8		8	8	8		6	6	4	
N	8	8	8			8	7	**9**	10	10	8		8	6	6		6	6		
O	6	5			6	5			7	6		**8**	8			**8**	6			**8**

Group totals: 82 | 86 | 121 | 98 | 74 — Total **407**

Block 4 (row id 7)

Row																				
L	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M	8	8	8		10	10	8		10	10	8		8	8	8		6	6	4	
N	8	8	8		8	8	8		10	10	8		8	6	6		6	6		
O	6	5			6	5			7	6		**8**	8			**8**	6			**8**

Group totals: 87 | 91 | 127 | 106 | 76 — Total **433**

Block 5 (row id 8)

Row																				
L	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M	8	8	8		10	10	8		10	10	8		8	8	8		6	6	4	
N	7	7	7		7	7	7		10	8	8	**9**	6	6	6	**9**	6	5		**9**
O	6				6				8				5				4			

Group totals: 79 | 83 | 122 | 103 | 75 — Total **405**

Block (9) — total 415

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L / LO	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M / MO	8	8	8		10	10	8		10	10	8		8	8	8		6	6	4	
N / NO	8	6	6			6	5	**9**	8	8	6	**9**	6	6	3	**9**	5	5		**9**
O / OO	6	5			6	5			7	6			8				6			

9   **83**   **87**   **123**   **103**   **76**

Block (10) — total 434

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L / LO	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M / MO	8	8	8		10	10	8		10	10	8		8	8	8		6	6	4	
N / NO	8	6	6		8	6	6		8	8	6	**9**	6	6	3	**9**	6	4		**9**
O / OO	6	6	3		6	6	3		6	6	5		7	6			8			

10   **87**   **91**   **127**   **108**   **78**

Block (11) — total 399

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L / LO	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M / MO	8	8	8		10	10	8		10	10	8		8	8	8		6	6	4	
N / NO	5	4			5	4			6	6		**9**	8			**9**	5			**9**
O / OO	6	6	4		6	6	4		8	6	6		6	6	3		6	4		

11   **77**   **81**   **121**   **104**   **76**

Block (12) — total 428

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L / LO	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M / MO	8	8	8			9	9	**10**	10	10	8		8	8	8		**20**	6	4	
N / NO	7	7			7	7			7	7		**9**	7			**9**	9			**9**
O / OO	6	6	4		6	6	4		8	6	6		6	6	3		6	4		

12   **82**   **86**   **123**   **103**   **94**

Block (13) — total 430

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L / LO	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M / MO	8	8	8		10	10	8		10	10	8		8	8	8		6	6	4	
N / NO	5	5	4		5	5	4		5	5	4	**9**	7			**9**	6			**9**
O / OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		

13   **86**   **90**   **127**   **108**   **79**

Block 14

L/LO	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M/MO	8	8	6		8	8	8	**10**	10	8	8	**10**	8	8	6	**10**	7	7		**10**
N/NO		8				8				8				5				4		
O/OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		
	78				**80**				**121**				**106**				**77**			

14 — 399

Block 15

L/LO	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M/MO	8	6	6			6	6	**10**	8	8	7	**10**	6	6	6	**10**	7	6		**10**
N/NO	5	5	4		5	5	4		5	5	4		7				6			
O/OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		
	82				**84**				**124**				**104**				**78**			

15 — 409

Block 16

L/LO	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M/MO	8	6	6		8	8	6		8	8	7	**10**	6	6	6	**10**	7	6		**10**
N/NO	6	6	6		6	6	6		8	6	6		6	6			8			
O/OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		
	86				**88**				**130**				**109**				**80**			

16 — 430

Block 17

L/LO	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M/MO	6	4			6	4			6	6		**10**	5	5		**10**	6			**10**
N/NO	8	6	6		8	6	6		8	8	6		6	6	3		5	5		
O/OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		
	78				**78**				**122**				**105**				**76**			

17 — 393

Block 18

L/LO	10	10	8			9	8	**11**	12	10	10		10	10	8		6	6	4	
M/MO	5	5	4		5	5	4		5	5	5	**10**	5	4		**10**	8			**10**
N/NO	8	6	6		8	6	6		8	8	6		6	6	3		5	5		
O/OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		
	82				**82**				**125**				**104**				**78**			

18 — 405

19

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L / LO	10	10	8		10	10	8		12	10	10		10	10	8		6	6	4	
M / MO	5	5	4		5	5	4		5	5	5	**10**	9			**10**	8			**10**
N / NO	8	8	8		8	8	8		10	10	8		8	6	6		6	6		
O / OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		

19   86   86   131   109   80   **426**

20

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L / LO	10	8	8		10	8	8		10	10	10	**11**	8	8	6	**11**	6	6	3	**11**
M / MO		8				8				9				6				5		
N / NO	8	8	8		8	8	8		10	10	8		8	6	6		6	6		
O / OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		

20   78   78   121   98   74   **392**

21

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L / LO	8	8	7			6	6	**11**	10	8	8	**11**	8	6	6	**11**	5	5	4	**11**
M / MO	7	7			7	7			8	7			6	3			4	4		
N / NO	8	8	8		8	8	8		10	10	8		8	6	6		6	6		
O / OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		

21   81   81   123   99   76   **403**

22

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L / LO	8	8	7		8	8	7		10	8	8	**11**	8	6	6	**11**	5	5	4	**11**
M / MO	6	6	6		8	6	6		8	6	6		6	6	3		6	5		
N / NO	8	8	8		8	8	8		10	10	8		8	6	6		6	6		
O / OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		

22   85   87   128   105   79   **427**

23

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L / LO	6	6			6	6			7	7		**11**	8			**11**	7			**11**
M / MO	8	6	6		8	8	6		8	8	7		6	6	6		7	6		
N / NO	8	8	8		8	8	8		10	10	8		8	6	6		6	6		
O / OO	8	6	6		8	6	6		8	8	8		8	6	6		6	6		

23   76   78   116   93   71   **386**

Table 11. Notional Detailed Daily Simulator Training Schedule

Daily Simulator Schedule

	Monday				Tuesday				Wednesday				Thursday				Friday			
L	4	4	4		4	4	4		3	3	2		2	2	2		2	2	2	
LO	2	2	2		2	2			2	2			2	2			2			
M	2	2			2	2			2	2			2	2			2			
MO																				
N																				
NO																				82
O																				
OO			22			20			16				14				10			

	Monday				Tuesday				Wednesday				Thursday				Friday			
L	4	4	4		4	4	4		3	3	2		2	2	2		2	2	2	
LO	2	2	2		2	2			2	2			2	2			2			
M	2	2			2	2			2	2			2	2			2			
MO																				
N																				
NO																				82
O																				
OO			22			20			16				14				10			

94

Chart 1 (total: 82)

Row															
L	4	4	4	4	4	4	3	3	2	2	2	2	2	2	2
LO	2	2	2	2	2		2	2		2	2		2		
M	2	2		2	2		2	2		2	2		2		
MO															
N															
NO															
O															

Axis (OO): 22, 20, 16, 14, 10

Chart 2 (total: 78)

Row															
L	3	3	3	3	3	3	2	2	2	2	2	1	2	2	
LO	2	2	2	2	2		2	2		2	2		2		
M	2			2			2			2			1		
MO	2	2		2	2		3			3			2		
N															
NO															
O															

Axis (OO): 21, 19, 15, 14, 9

Chart 3 (total: 78)

Row															
L	3	3	3	3	3	3	2	2	2	2	2	1	2	2	
LO	2	2	2	2	2		2	2		2	2		2		
M	2			2			2			2			1		
MO	2	2		2	2		3			3			2		
N															
NO															
O															

Axis (OO): 21, 19, 15, 14, 9

Chart 4 (total: 78)

Row															
L	3	3	3	3	3	3	2	2	2	2	2	1	2	2	
LO	2	2	2	2	2		2	2		2	2		2		
M	2			2			2			2			1		
MO	2	2		2	2		3			3			2		
N															
NO															
O															

Axis (OO): 21, 19, 15, 14, 9

Chart 5 (total: 74)

Row															
L	2	2	2	2	2	2	2	2		2	2		2		
LO	2	2	2	2	2		2	2		2	2		2		
M															
MO	3	3	2	3	3	2	2	2	2	2	2	2	2	2	
N															
NO															
O															

Axis (OO): 20, 18, 14, 14, 8

Grid 1

																74
L	2	2	2		2	2	2		2	2			2	2			2			
LO	2	2	2		2	2			2	2			2	2			2			
M																				
MO	3	3	2		3	3	2		2	2	2		2	2	2		2	2		
N																				
NO																				
O																				
OO		20				18				14				14				8		

Grid 2

																74
L	2	2	2		2	2	2		2	2			2	2			2			
LO	2	2	2		2	2			2	2			2	2			2			
M																				
MO	3	3	2		3	3	2		2	2	2		2	2	2		2	2		
N																				
NO																				
O																				
OO		20				18				14				14				8		

Grid 3

																74
L	2	2	2		2	2	2		2	2			2	2			2			
LO	3				2				2				2				1			
M	2	2			2	2			2	2			3				3			
MO	2	2	2		2	2	2		2	2	1		2	2	1		2			
N																				
NO																				
O																				
OO		19				18				15				14				8		

Grid 4

																74
L	2	2	2		2	2	2		2	2			2	2			2			
LO	3				2				2				2				1			
M	2	2			2	2			2	2			3				3			
MO	2	2	2		2	2	2		2	2	1		2	2	1		2			
N																				
NO																				
O																				
OO		19				18				15				14				8		

Grid 5

																74
L	2	2	2		2	2	2		2	2			2	2			2			
LO	3				2				2				2				1			
M	2	2			2	2			2	2			3				3			
MO	2	2	2		2	2	2		2	2	1		2	2	1		2			
N																				
NO																				
O																				
OO		19				18				15				14				8		

Block 1:

L	2	2	2		2	2	2		2	2			2	2			2			
LO																				
M	3	3	2		3	3	2		3	3	2		2	2	2		2	2	2	
MO	2	2			2	2			2	2			2	2						
N																				
NO																				
O																				

OO   18   18   16   14   8   74

Block 2:

L	2	2	2		2	2	2		2	2			2	2			2			
LO																				
M	3	3	2		3	3	2		3	3	2		2	2	2		2	2	2	
MO	2	2			2	2			2	2			2	2						
N																				
NO																				
O																				

OO   18   18   16   14   8   74

Block 3:

L	2	2	2		2	2	2		2	2			2	2			2			
LO																				
M	3	3	2		3	3	2		3	3	2		2	2	2		2	2	2	
MO	2	2			2	2			2	2			2	2						
N																				
NO																				
O																				

OO   18   18   16   14   8   74

Block 4:

L	2	2	2		2	2	2		2	2			2	2			2			
LO																				
M	3	3	2		3	3	2		3	3	2		2	2	2		2	2	2	
MO	2	2			2	2			2	2			2	2						
N																				
NO																				
O																				

OO   18   18   16   14   8   74

Block 5:

L	2	2	2		2	2	2		2	2			2	2			2			
LO																				
M	3	3	2		3	3	2		3	3	2		2	2	2		2	2	2	
MO	2	2			2	2			2	2			2	2						
N																				
NO																				
O																				

OO   18   18   16   14   8   74

Grid 1 (total: 74)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L	2	2	2		2	2	2		2	2			2	2			2			
LO																				
M	3	3	2		3	3	2		3	3	2		2	2	2		2	2	2	
MO	2	2			2	2			2	2			2	2						
N																				
NO																				
O																				
OO		18				18				16				14				8		

Grid 2 (total: 74)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L																				
LO	4	4	4		3	3	2		3	3	2		2	2	2		2	2	2	
M	2	2			2	2			2	2			2	2			2			
MO	2	2			2	2			2	2			2	2						
N																				
NO																				
O																				
OO		20				16				16				14				8		

Grid 3 (total: 74)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L																				
LO	4	4	4		3	3	2		3	3	2		2	2	2		2	2	2	
M	2	2			2	2			2	2			2	2			2			
MO	2	2			2	2			2	2			2	2						
N																				
NO																				
O																				
OO		20				16				16				14				8		

Grid 4 (total: 74)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L																				
LO	4	4	4		3	3	2		3	3	2		2	2	2		2	2	2	
M	2	2			2	2			2	2			2	2			2			
MO	2	2			2	2			2	2			2	2						
N																				
NO																				
O																				
OO		20				16				16				14				8		

Grid 5 (total: 78)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
L	2	2	2		2	2	2		2	2			3				3			
LO	3	3	3		2	2	2		2	2	2		2	2	1		2	2		
M	2	2			2	2			2	2			2	2			2			
MO	2				2				2				2							
N																				
NO																				
O																				
OO		21				18				16				14				9		

Table 1

L	2	2	2		2	2	2		2	2			3				3		
LO	3	3	3		2	2	2		2	2	2		2	2	1		2	2	
M	2	2			2	2			2	2			2	2			2		
MO	2				2				2				2						
N																			
NO																			
O																			

OO   21   18   16   14   9   78

Table 2

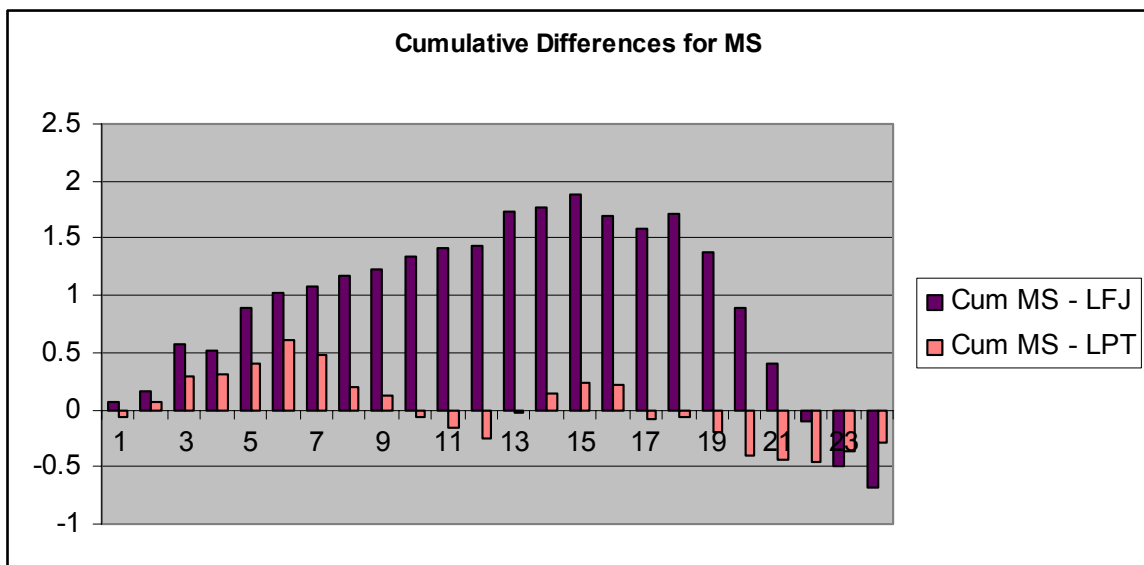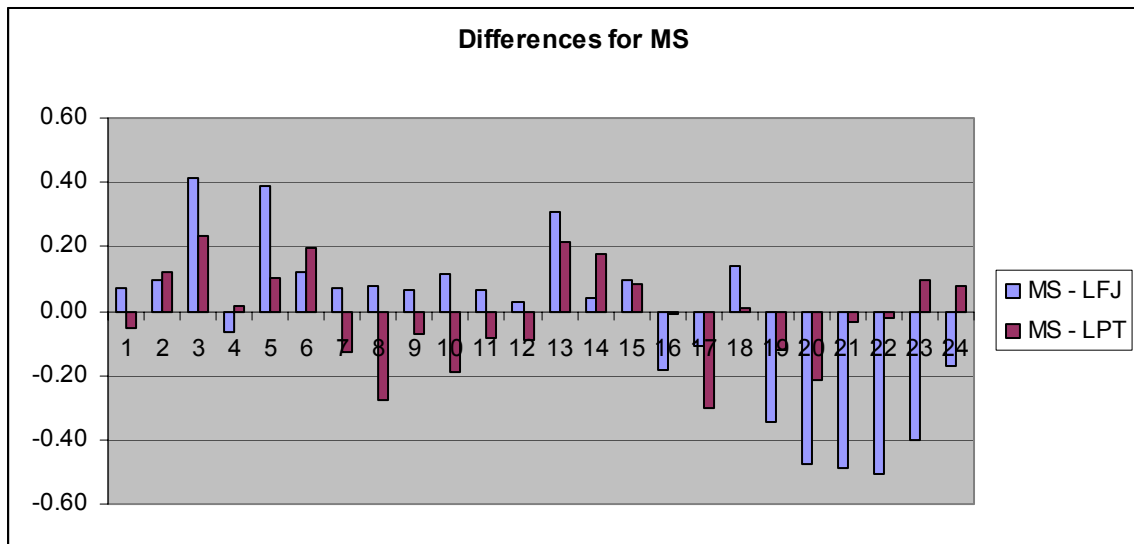| L	2	2	2		2	2	2		2	2			3				3		
LO	3	3	3		2	2	2		2	2	2		2	2	1		2	2	
M	2	2			2	2			2	2			2	2			2		
MO	2				2				2				2						
N																			
NO																			
O																			

OO   21   18   16   14   9   78

Appendix C: Extra Charts

This appendix contains the extra charts for the calculations of the difference between the various scheduling rules. The charts were not included in the body.

Differences for LPT

Cumulative Difference for LPT

Differences for MS

Cumulative Differences for MS

102
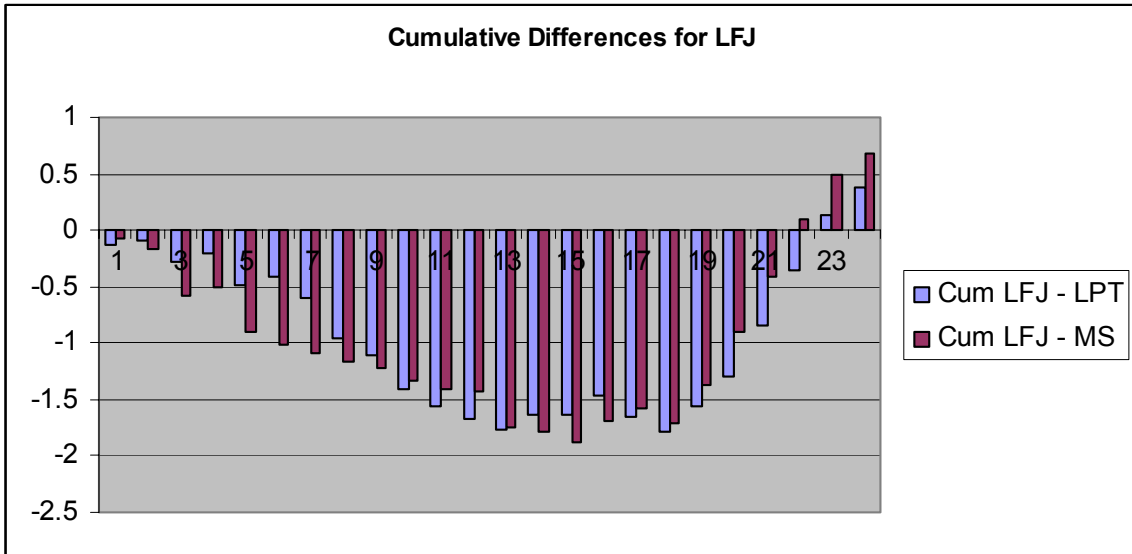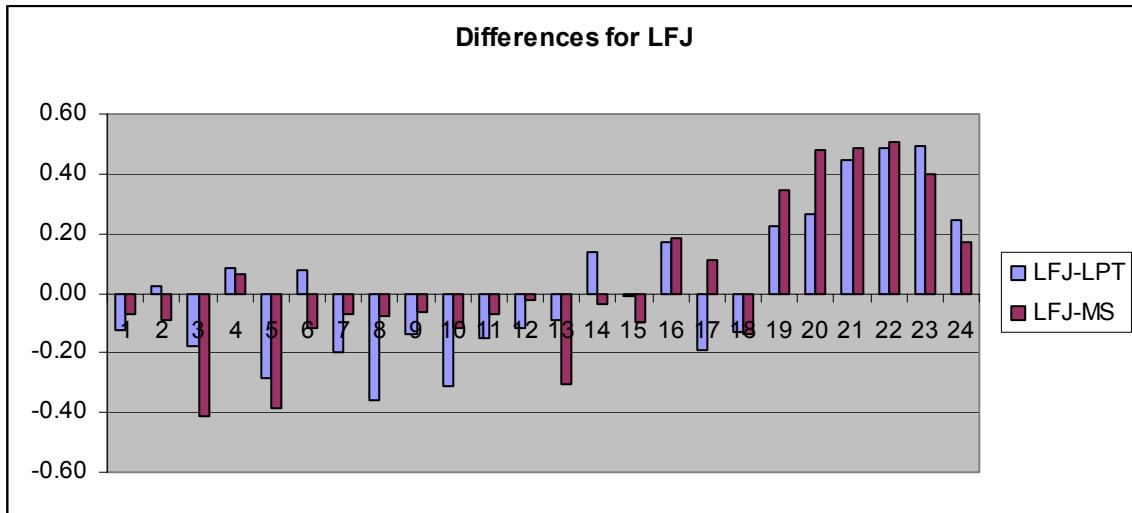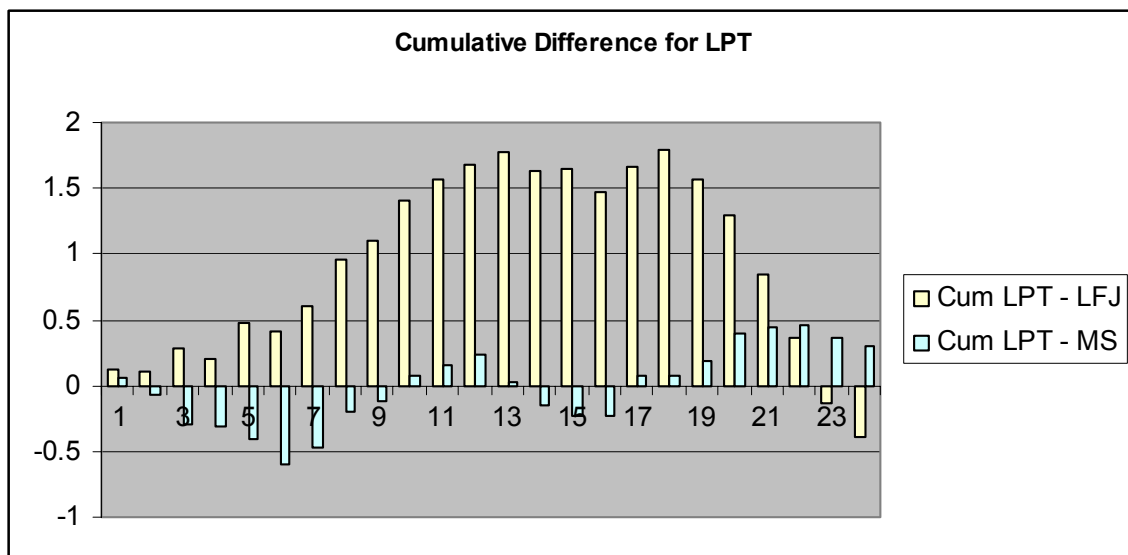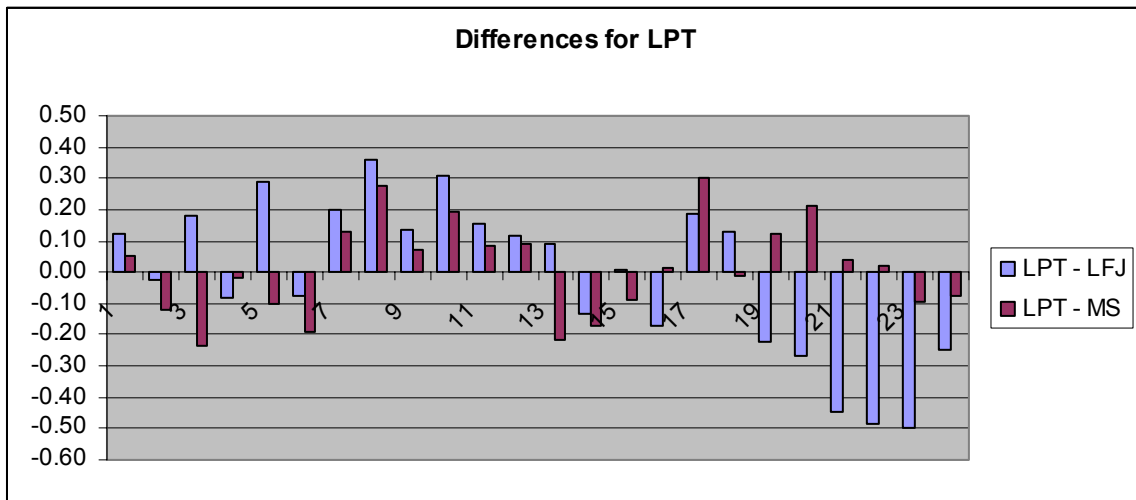
Bibliography

1. "87th Flying Training Squadron Scheduling Procedures." April 2000.

2. "87th FTS Operating Instructions 11-201." *Squadron Policies and Procedures*. 12 August 2000.

3. Air Education Training Command. "T-38 Specialized Undergraduate Pilot Training Syllabus," *AETC Syllabus P-V4A-A (T-38)*, June 2001.

4. Belton, Valerie, and Mark Elder. "Exploring a Multicriteria Approach to Production Scheduling," *Journal of the Operations Research Society*, 1996, 47:162-174.

5. Calhoun, Kevin Capt. Phone Interview and Correspondence, 87th Flying Training Squadron, Laughlin AFB TX.

6. Chu, Sydney C.K., "A Goal Programming Model for Crew Duties Generation." *Journal of Multi-Criteria Decision Analysis*, 2001, 10:143-151.

7. Deitel, H.M. and Deitel, P.J. *C: How to Program*, 2nd ed., Prentice Hall, 1994.

8. Deitel, H.M. and Deitel, P.J. *How to Program Java,* 3rd ed. Prentice Hall, 1999.

9. Desrosiers, Jacques, Daniel McInnis, Arielle Lasry, Marius M. Solomon, and Francois Soumis. "Air Transat Uses ALTITUDE to Manage Its Aircraft Routing, Crew Pairing, and Work Assignment," *Interfaces*, March-April 2000, Vol 30:41-53.

10. Dupont, Noel J., Nguyen, Cuong T., Pektas, Mustafa K. "Comparative of Computational Languages: Visual Basic vs. C++ vs. Java," AFIT OPER 699 Class Project Paper, Fall 2001, 39 pages.

11. Evren, Fuat, SSS: "A Knowledge-Based Approach to Resource Scheduling in an F-16 Fighter Training Unit." Term Project, Middle East Technical University, 1999.

12. Housos, Efthymiou, and Tony Elmroth. "Automatic Optimization of Subproblems in Scheduling Airline Crews," *Interfaces*, 1997, Vol. 27, 5:68-77.

13. Kiely, Don. "Visual Basic App Edition Eases Code Reuse," *Information Week*, 1997, Issue 656:10-12.

14. Kiely, Don. "License to Integrate," Information Week, 1997, Issue 658:

15. Kiely, Don. "Microsoft Takes New Direction with VBA 6.0," *Information Week*, May 31, 1999. p.8a.

16. Kim, Eugene E. "Programming & The PC Revolution," *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, Jan 2001, Vol 26, Issue1:21-28.

17. Klamroth, Kathrin, and Margaret M. Wiecek. "A Time-Dependent Multiple Criteria Single-Machine Scheduling Problem." *European Journal of Operations Research*, 2001, 135:17-26.

18. Klein, Yaron, and Gideon Langholz. "Multi-Criteria Scheduling Optimization Using Fuzzy Logic." Department of Electrical Engineering – Systems, Tel-Aviv University.

19. Korhonen, Pekka, Seppo Salo, and Ralph E. Steuer. "A Heuristic For Estimating Nadir Criterion Values in Multiple Objective Linear Programming." *Operations Research*, Sep-Oct 1997, Vol.45 I5:751-757.

20. McCurdy, Richard "Cheese." Initial Meeting and Interview, 13-15 September 2001, 87th Flying Training Squadron, Laughlin AFB TX.

21. Noronha, S.J., and V.V.S. Sarma. "Knowledge-based approaches for scheduling problems: a survey." *IEEE Transactions on Knowledge and Data Engineering*, 1991; 3:160-171.

22. Rakshit, Ananda, Nirup Krishnamurthy, and Gang Yu. "System Operations Advisor: A Real-Time Decision Support System for Managing Airline Operations at United Airlines." *Interfaces*, 1996, Vol. 26, I2:50-58.

23. Raytheon Aircraft Company, *Software User's Manual for the Training Integrated Management System (TIMS) of the Joint Primary Aircraft Training System (JPATS) Ground Based Training System (GBTS), Volume I, II, and III*, November 19, 2001. pp. 1-1191.

24. Steuer, Ralph E. *Multiple Criteria Optimization: Theory, Computation, and Application*, John Wiley & Sons, 1986.

25. Stojkovic M, Soumis F, Desrosiers J. "The Operational Airline Crew Scheduling Problem," *Transportation Science*, Vol. 32, No. 3, August 1998.

26. Vance, Pamela H., Cynthia Barnhart, Ellis J. Johnson, and George L. Nemhauser. "Airline Crew Scheduling: A New Formulation and Decomposition Algorithm," *Operations Research*, 1997, Vol. 45, 2:188-200.

27. Yager, Tom. "Windows NT C/C++ Development Basics, Part I," *Performance Computing*, April 98, Vol.1, I1:33-36.

28. Yan, Shangyao, and Jei-Chi Chang, "Airline Cockpit Crew Scheduling," *European Journal of Operations Research*, 2002, 136:501-511.

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
26-03-2002	Master's Thesis	May 2001 – Mar 2002

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
AN INTERACTIVE DECISION SUPPORT SYSTEM FOR SCHEDULING FIGHTER PILOT TRAINING	
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Nguyen, Cuong T., Captain, USAF	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	8. PERFORMING ORGANIZATION REPORT NUMBER
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640, WPAFB OH 45433-7765	AFIT/GOR/ENS/02-12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
87th Flying Training Squadron, Public Affairs 561 Liberty Drive, Suite 3, Laughlin AFB, Texas 78843-5227	
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Fighter Pilots students undertake an intense 120-day training program. New classes of students enter the training program at regular interval. Students endured rigorous academic, simulator, and aircraft training throughout the program. Squadron schedulers ensure the multiple classes and students are scheduled for the activities. Simulator and aircraft training are scheduled individual for each student. Academic training are taught to the class. Aircraft utilization must also be considered. Aircraft Sortie training are also constrained by daylight hours. Additionally, students are limited to a maximum of three training events in a given day. Squadron schedulers must balance these requirements to ensure students meet their training requirements and successfully graduate. The dynamic training environment requires advanced robust schedules with flexibility to accommodate changes.

A Visual Interactive Modeling approach is used to generate schedules. Current schedules are being generated manually with an Excel spreadsheet. Taking advantage of Excel's Visual Basic programming language, the Excel tool is modified in several ways. Scheduling Dispatch rules are implemented to automatically generate feasible schedules. Graphical User Interfaces are used to create a user-friendly environment. Schedulers guide the schedule building process to produce a robust schedule. An attrition environment is created to simulate attrition probabilities of aircraft sortie training due to operations, maintenance, weather, and other cancellations. Analysis of dispatch rules are analyzed.

15. SUBJECT TERMS Scheduling, Fighter Pilot Scheduling, Excel VBA Applications

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Richard F Deckro, Professor, AFIT
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	116	19b. TELEPHONE NUMBER *(Include area code)*
U	U	U			(937) 255-6565, ext 4325 (Richard.Deckro@afit.edu)